AMS infoAdvantage

AMS infoAdvantage Designer

Presented by CGI-AMS

July 20, 2005

Agenda

- Designer Overview
- Data Warehouse
- Delivered Universes
- Universe Development
  - Adding Tables
  - Table Joins
  - Objects and Classes

Business Object Tools

- Supervisor
  - Security access
- **Designer**
  - Create/Maintain Universes
    - Universe: Grouping of tables. Defines what fields can be used in reports
- Business Objects Reports viewer/creator:
  - ‘Thick Client’
  - Create/run complex reports
  - Installed on your machine
- WebIntelligence:
  - ‘Thin Client’ or WebI
  - Run any kind of report
  - Online reporting piece

Designer

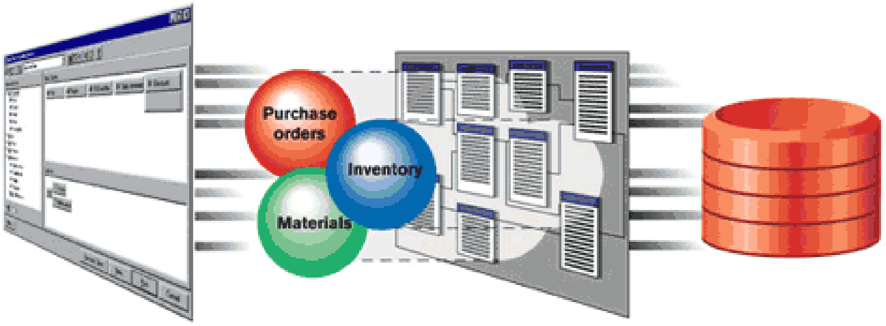
- Create universes
- Modify universes
  - Add/Remove Tables
  - Add/Remove Fields
  - Change Database connection
  - Define reporting parameters
- Mostly used by Systems Administrator

Building Universes

- A good universe:
  - Gives access to the data that is relevant to the users it is designed for
  - Is easy to use
  - Generates efficient SQL
- General rule: smaller is better
- AMS infoAdvantage approach: universes based on functional area

Universes – revisit

- Layer between end user and reporting database
- End user does not need to know SQL
- Fields are described in every day English
- Universes contain a subset of the table structure used for reporting:
  - Database Tables, Fields
  - Joins to link tables

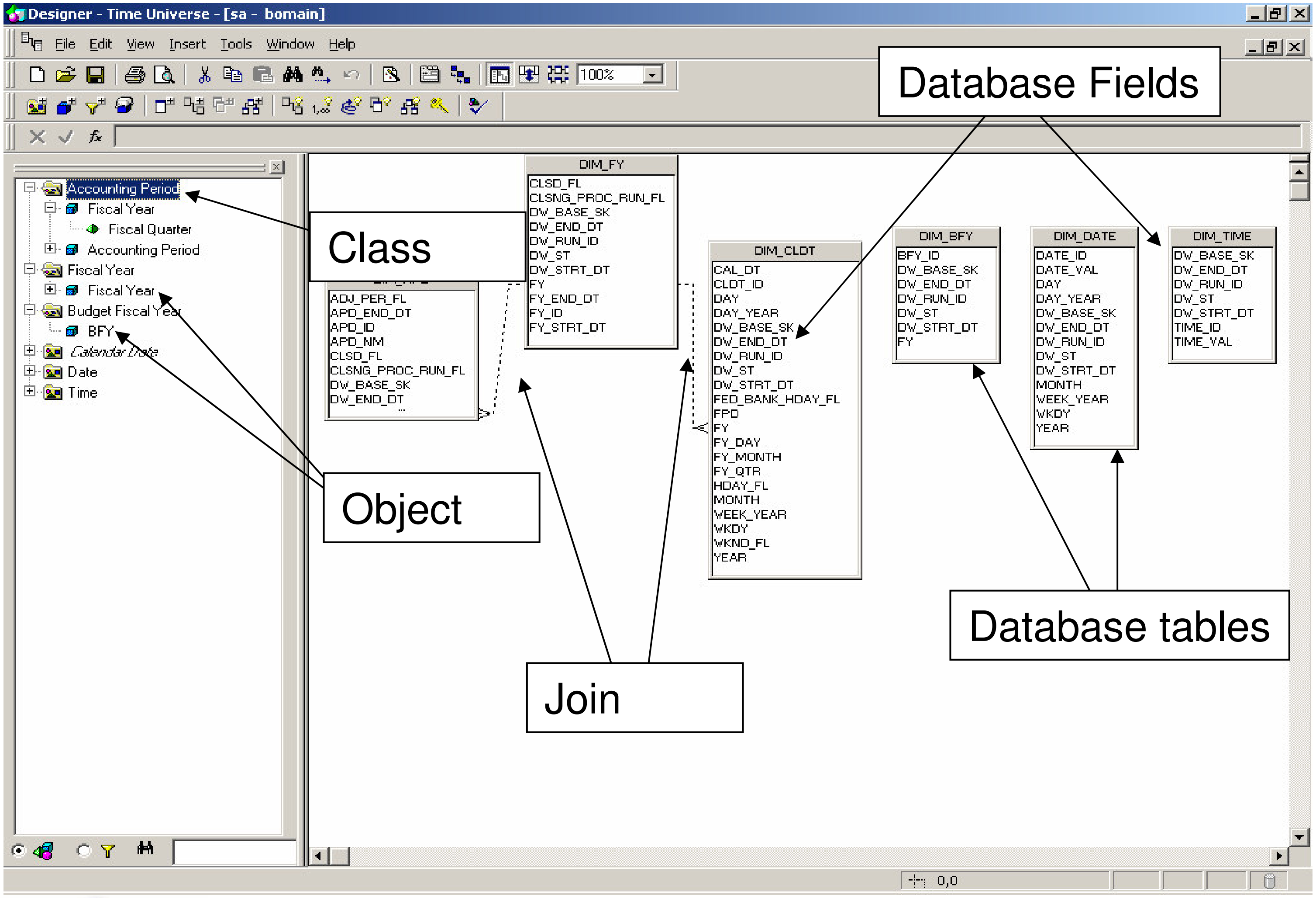


- Universes often created around functional areas
- Consist of classes and objects
- Access data in the data warehouse through universes

Universes

- A universe is a layer of abstraction between the user and the reporting database. The Designer includes tables and joins in the universe, while the user creates reports without having to know SQL.
- The role of a universe is to provide an easy to use and understand interface for non technical Business Objects and WebIntelligence users to run queries against a database to create reports and perform data analysis.
- Business Objects stores data in the Repository database. The universe must be exported to the repository for a Web User to see it

Universe - Example



Designer - Time Universe - [sa - bomain]
File Edit View Insert Tools Window Help
100%
0,0

- Accounting Period
  - Fiscal Year
    - Fiscal Quarter
  - Accounting Period
- Fiscal Year
  - Fiscal Year
- Budget Fiscal Year
  - BFY
- *Calendar Date*
- Date
- Time

DIM_FY
CLSD_FL
CLSNG_PROC_RUN_FL
DW_BASE_SK
DW_END_DT
DW_RUN_ID
DW_ST
DW_STRT_DT
FY
FY_END_DT
FY_ID
FY_STRT_DT

ADJ_PER_FL
APD_END_DT
APD_ID
APD_NM
CLSD_FL
CLSNG_PROC_RUN_FL
DW_BASE_SK
DW_END_DT
...

DIM_CLDT
CAL_DT
CLDT_ID
DAY
DAY_YEAR
DW_BASE_SK
DW_END_DT
DW_RUN_ID
DW_ST
DW_STRT_DT
FED_BANK_HDAY_FL
FPD
FY
FY_DAY
FY_MONTH
FY_QTR
HDAY_FL
MONTH
WEEK_YEAR
WKDY
WKND_FL
YEAR

DIM_BFY
BFY_ID
DW_BASE_SK
DW_END_DT
DW_RUN_ID
DW_ST
DW_STRT_DT
FY

DIM_DATE
DATE_ID
DATE_VAL
DAY
DAY_YEAR
DW_BASE_SK
DW_END_DT
DW_RUN_ID
DW_ST
DW_STRT_DT
MONTH
WEEK_YEAR
WKDY
YEAR

DIM_TIME
DW_BASE_SK
DW_END_DT
DW_RUN_ID
DW_ST
DW_STRT_DT
TIME_ID
TIME_VAL

Database Fields

Class

Object

Join

Database tables

Universes

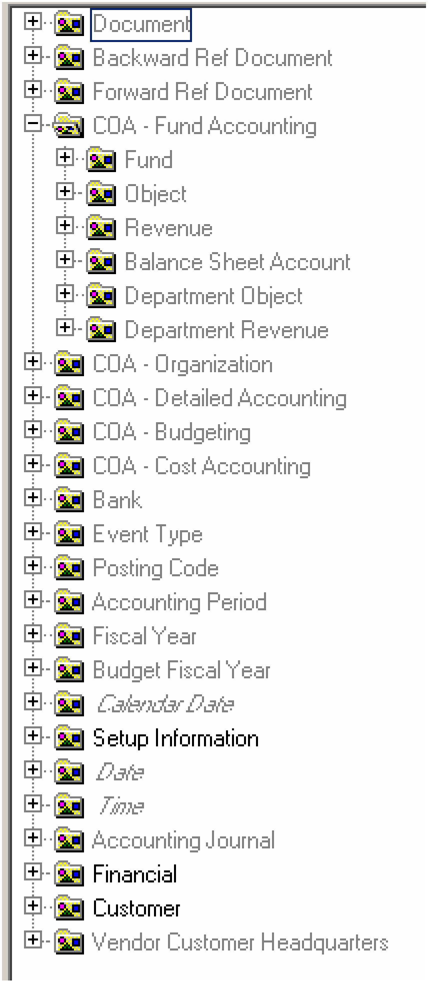
- Universe is divided into two panels
  - Right Panel (Structure Pane): Database tables and relationships; End Users cannot see the Right Panel when creating a report
  - Left Panel (Universe Pane): Defines the Business Friendly Classes and Objects seen by an End User during report creation; Each Object refers to a database field from a table added in the Right Panel

Universes – classes and objects

- Classes and Objects are equivalent to tables and rows
  - Object: component that maps to data in the database
    - Dimension: Typically a character field; it is the default
    - Measure: Numeric fields; can be aggregated
    - Detail: Description of Dimension
  - Classes: groups of objects

Class

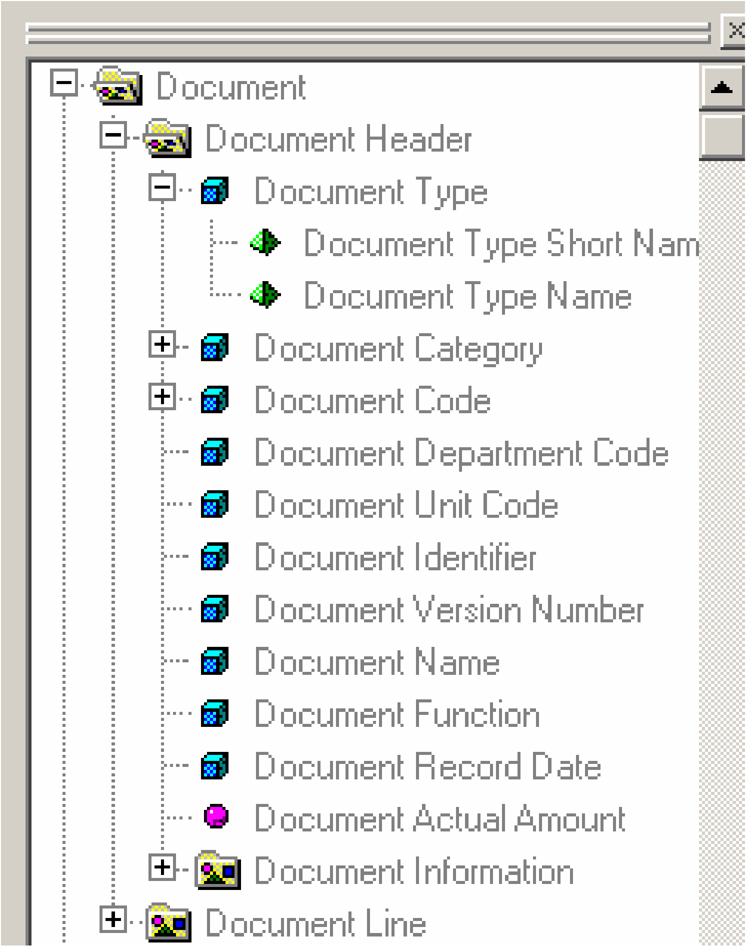
- Objects are grouped into Classes
- Provides a structure for the universe so that it is easier for the users to locate particular objects
- Sub-classes can be used to break down a set of related objects



- Document
- Backward Ref Document
- Forward Ref Document
- COA - Fund Accounting
  - Fund
  - Object
  - Revenue
  - Balance Sheet Account
  - Department Object
  - Department Revenue
- COA - Organization
- COA - Detailed Accounting
- COA - Budgeting
- COA - Cost Accounting
- Bank
- Event Type
- Posting Code
- Accounting Period
- Fiscal Year
- Budget Fiscal Year
- *Calendar Date*
- Setup Information
- *Date*
- *Time*
- Accounting Journal
- Financial
- Customer
- Vendor Customer Headquarters

Objects

- Dimension – Projects columns from the database which are key to a query
- Detail – Projects columns from the database that provide detailed information related to a dimension
- Measure – Contains aggregates to project statistics



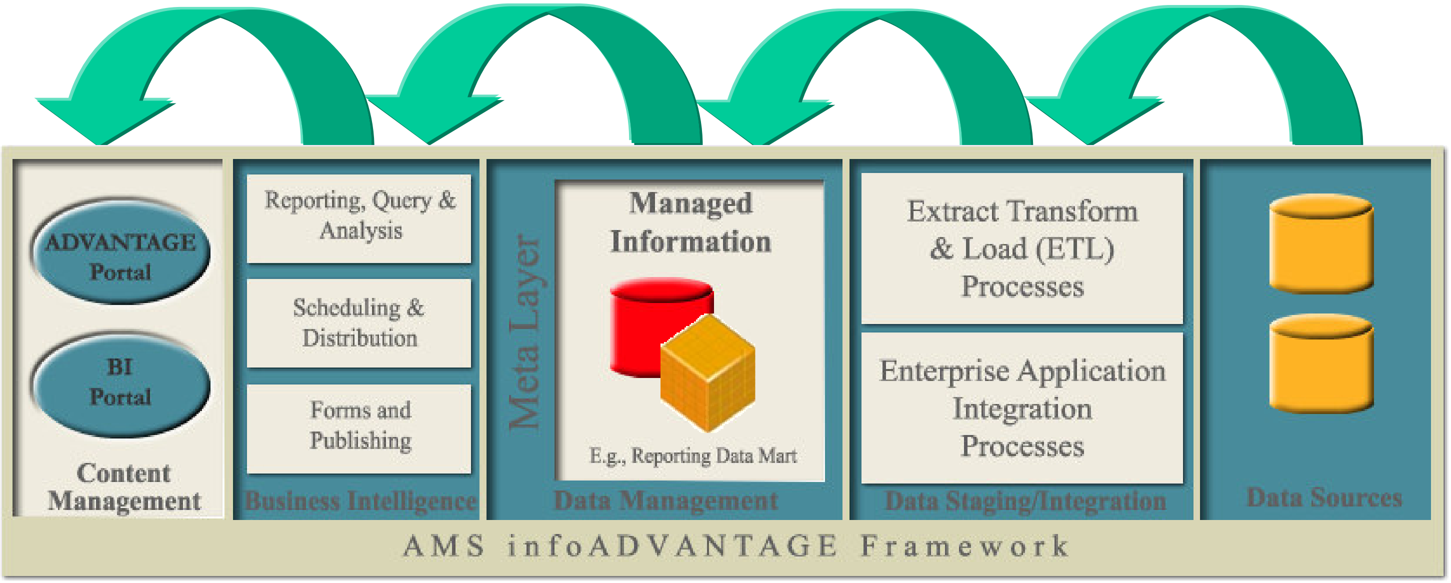
- Document
  - Document Header
    - Document Type
      - Document Type Short Nam
      - Document Type Name
    - Document Category
    - Document Code
    - Document Department Code
    - Document Unit Code
    - Document Identifier
    - Document Version Number
    - Document Name
    - Document Function
    - Document Record Date
    - Document Actual Amount
    - Document Information
  - Document Line

Structure of Classes and Objects

- Dimension objects tend to be organized hierarchically within a class
- Detail objects hang off dimension objects
- Measure objects are usually kept in a separate class, unless it is needed to be emphasized that they should be used with specific objects

3.4 infoAdvantage Universes

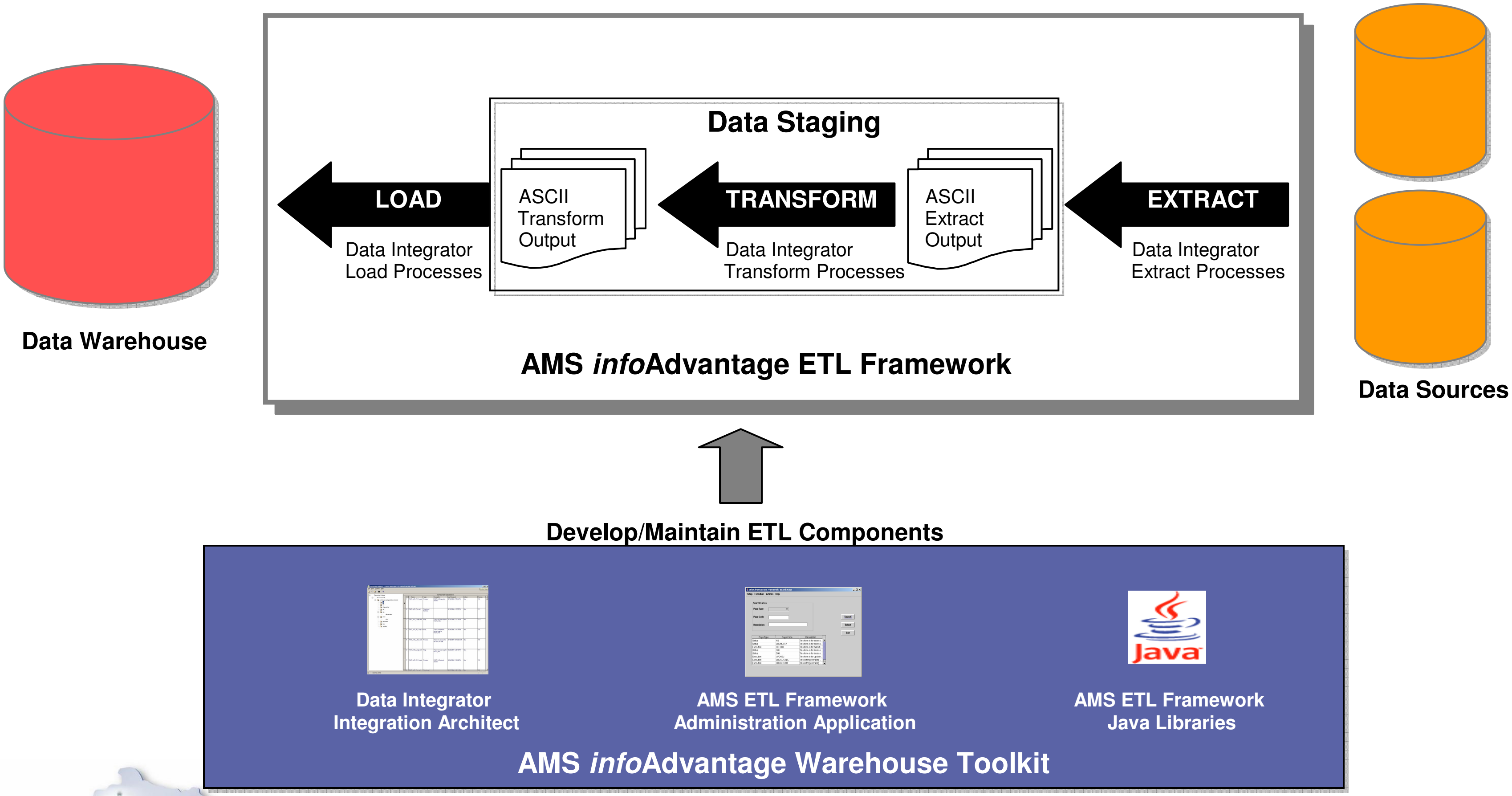
AMS *info*Advantage Framework


ADVANTAGE Portal
BI Portal
Content Management
Reporting, Query & Analysis
Scheduling & Distribution
Forms and Publishing
Business Intelligence
Meta Layer
Managed Information
E.g., Reporting Data Mart
Data Management
Extract Transform & Load (ETL) Processes
Enterprise Application Integration Processes
Data Staging/Integration
Data Sources
AMS infoADVANTAGE Framework

The Flow of Data into Information

- Data Source
- Data Staging and Integration
- Data Management
- Business Intelligence and Content Management

infoAdvantage Extract Transform Load Process



Data Warehouse

AMS *info*Advantage ETL Framework

Data Staging

LOAD

Data Integrator
Load Processes

ASCII
Transform
Output

TRANSFORM

Data Integrator
Transform Processes

ASCII
Extract
Output

EXTRACT

Data Integrator
Extract Processes

Data Sources

Develop/Maintain ETL Components

Data Integrator
Integration Architect

AMS ETL Framework
Administration Application

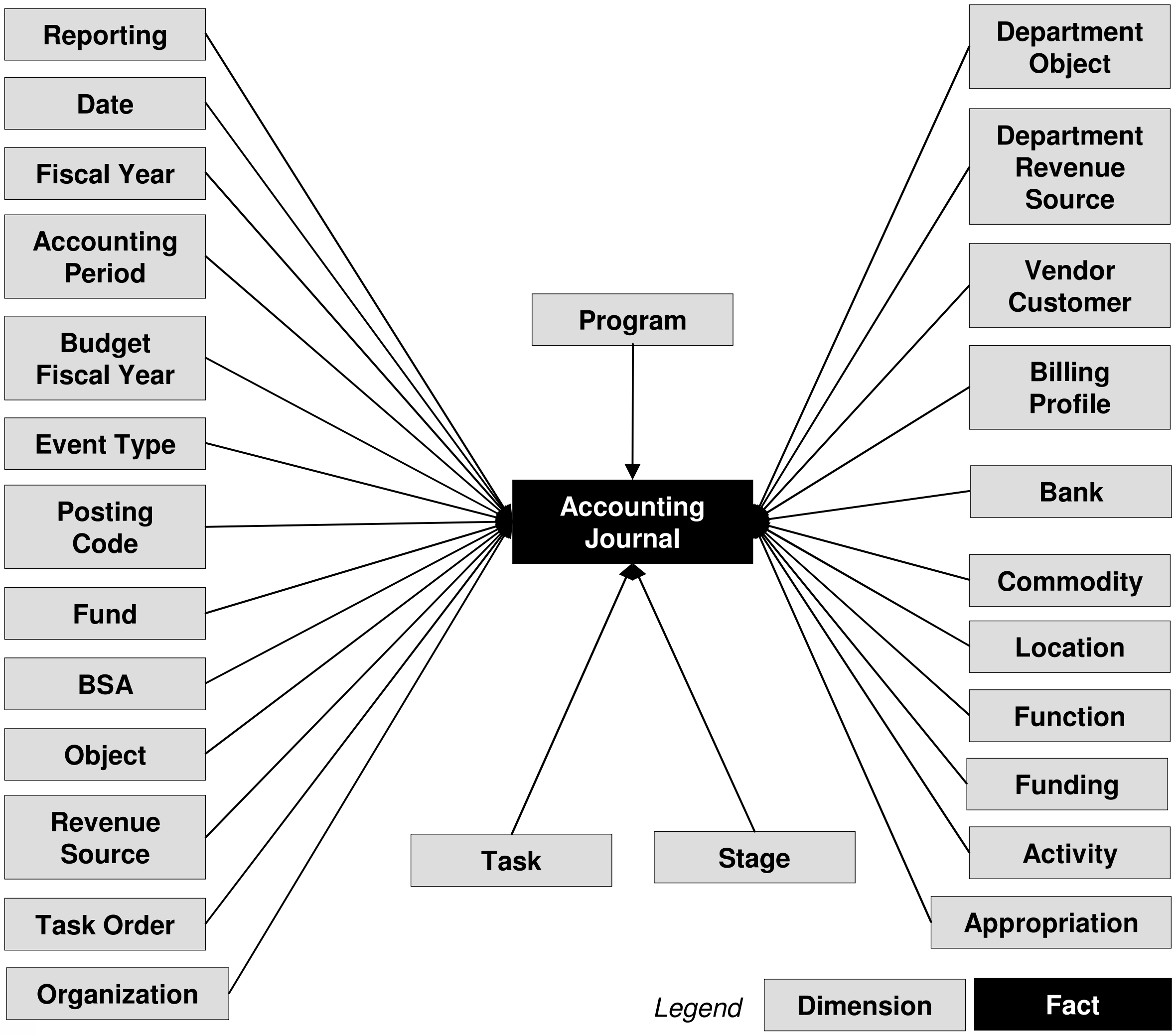
AMS ETL Framework
Java Libraries

AMS *info*Advantage Warehouse Toolkit

Universes

- The infoAdvantage datawarehouse has dimensions and facts in the star schema design
- Fact: A table which stores factual data or numeric values for analysis and aggregation
  - Example: Accounting Journal
- Dimension: A table that stores information describing the factual data and provides analysts with a way to sort, group and filter the data
  - Example: Fund

AMS infoAdvantage 3.4 GA Data Model



Reporting

Date

Fiscal Year

Accounting Period

Budget Fiscal Year

Event Type

Posting Code

Fund

BSA

Object

Revenue Source

Task Order

Organization

Program

Accounting Journal

Task

Stage

Department Object

Department Revenue Source

Vendor Customer

Billing Profile

Bank

Commodity

Location

Function

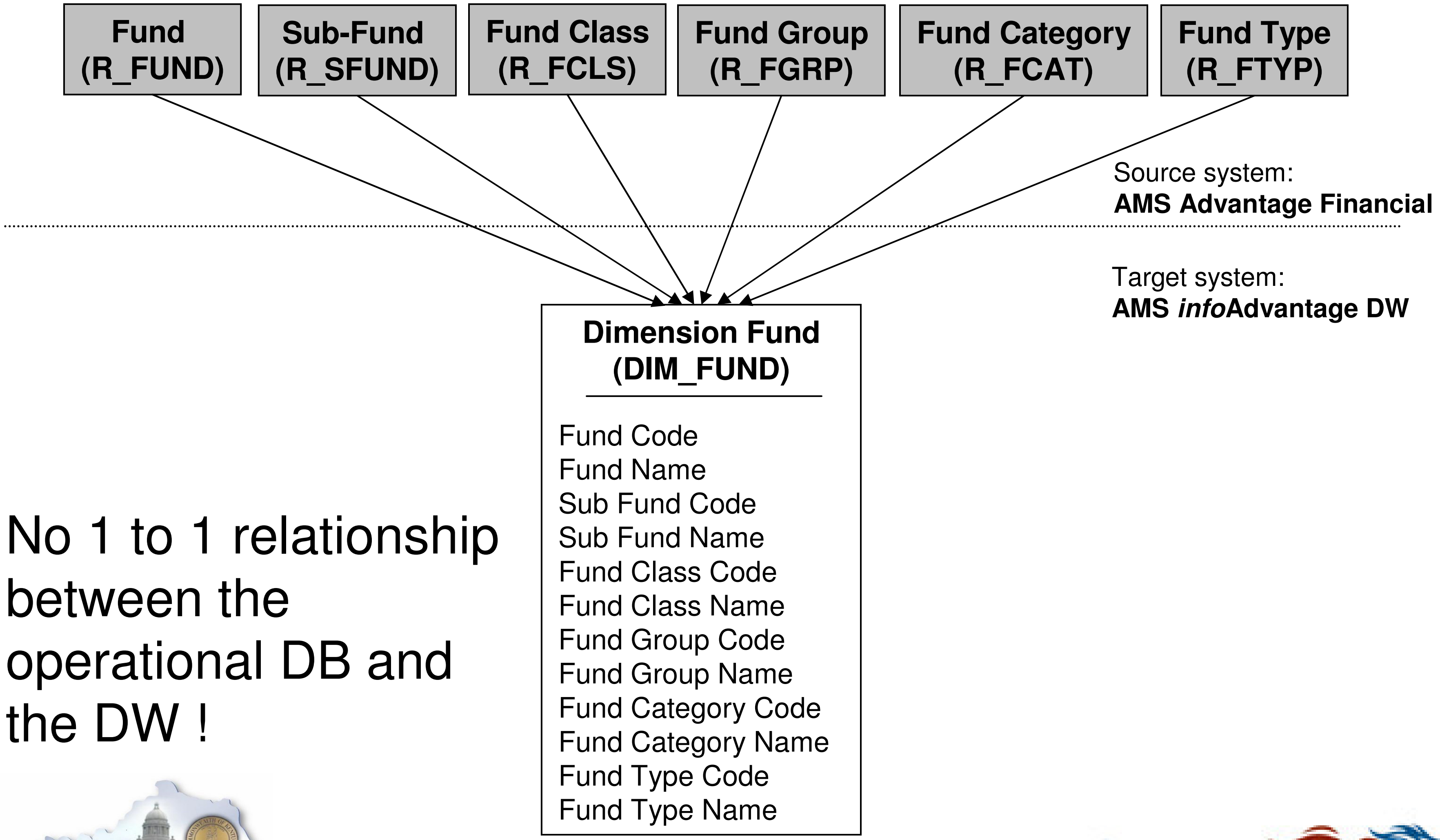
Funding

Activity

Appropriation

Legend Dimension Fact

AMS infoAdvantage 3.4 Dimension



Fund (R_FUND)

Sub-Fund (R_SFUND)

Fund Class (R_FCLS)

Fund Group (R_FGRP)

Fund Category (R_FCAT)

Fund Type (R_FTYP)

Source system:
AMS Advantage Financial

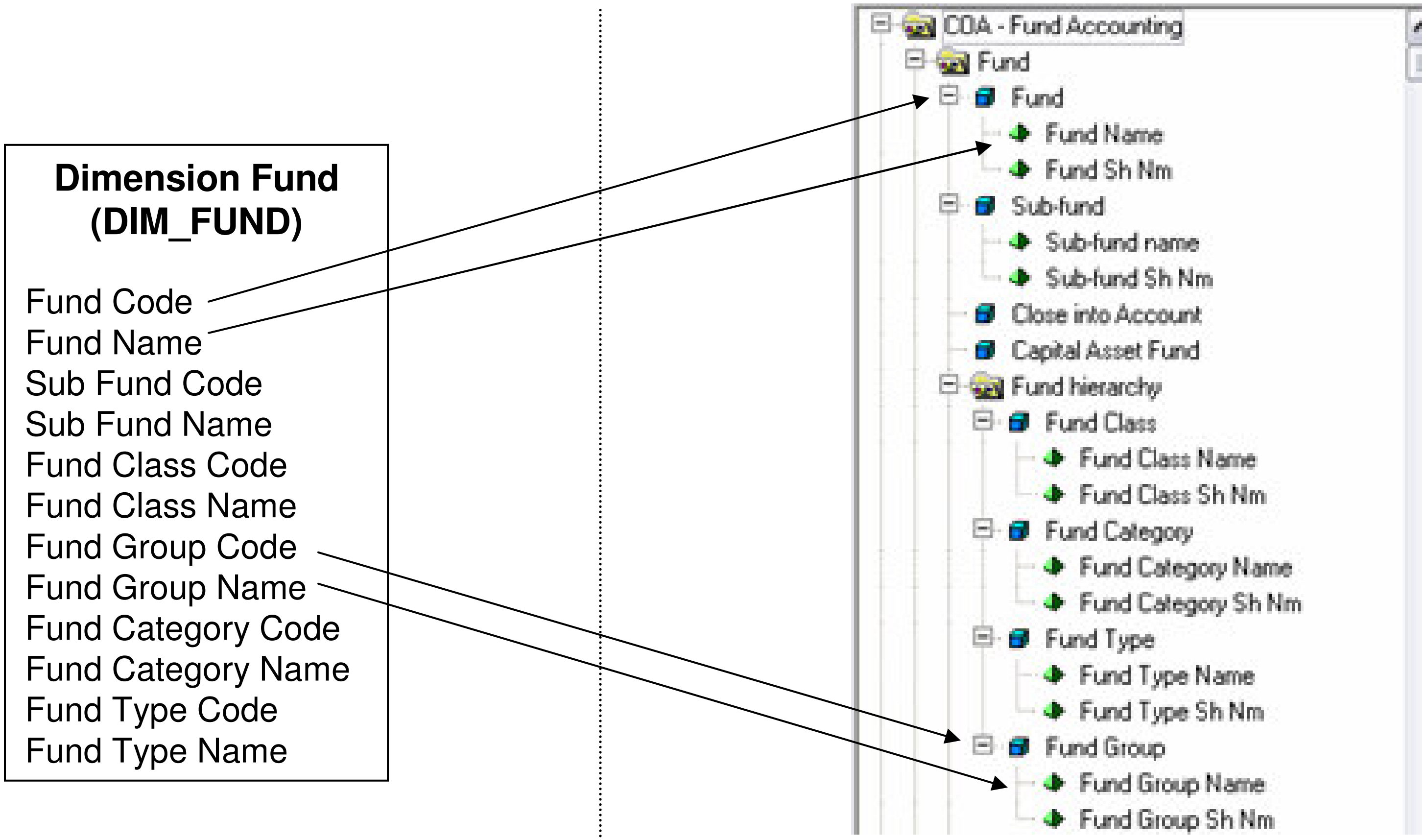
Target system:
AMS *info*Advantage DW

Dimension Fund (DIM_FUND)

Fund Code
Fund Name
Sub Fund Code
Sub Fund Name
Fund Class Code
Fund Class Name
Fund Group Code
Fund Group Name
Fund Category Code
Fund Category Name
Fund Type Code
Fund Type Name

No 1 to 1 relationship between the operational DB and the DW !

AMS infoAdvantage 3.4 Universe



Dimension Fund (DIM_FUND)

Fund Code
Fund Name
Sub Fund Code
Sub Fund Name
Fund Class Code
Fund Class Name
Fund Group Code
Fund Group Name
Fund Category Code
Fund Category Name
Fund Type Code
Fund Type Name

- COA - Fund Accounting
  - Fund
    - Fund
      - Fund Name
      - Fund Sh Nm
    - Sub-fund
      - Sub-fund name
      - Sub-fund Sh Nm
    - Close into Account
    - Capital Asset Fund
    - Fund hierarchy
      - Fund Class
        - Fund Class Name
        - Fund Class Sh Nm
      - Fund Category
        - Fund Category Name
        - Fund Category Sh Nm
      - Fund Type
        - Fund Type Name
        - Fund Type Sh Nm
      - Fund Group
        - Fund Group Name
        - Fund Group Sh Nm

Data Layer Level (database)

Presentation Layer Level (universe)

Delivered Universes

- **Chart of Accounts Universe** – includes all chart of account elements
- **Time Universe** – provides information about fiscal year, accounting period etc.
- **Geographic Universe** – provides information about state, county etc.
- **Vendor/Customer Universe** – includes information on vendors
- **Accounting Journal Universe** – includes accounting journal and cost accounting journal information
- **Document Catalog Universe** – provides information on processed documents
- **General Accounting Universe** – includes information on financial transactions
- **Cost Accounting Universe** – provides information on accounting events
- **Accounts Receivable Universe** –shows information on transactions related to the collection of money owed
- **Budget Execution Universe** – includes information budget establishment and execution

Delivered Reports

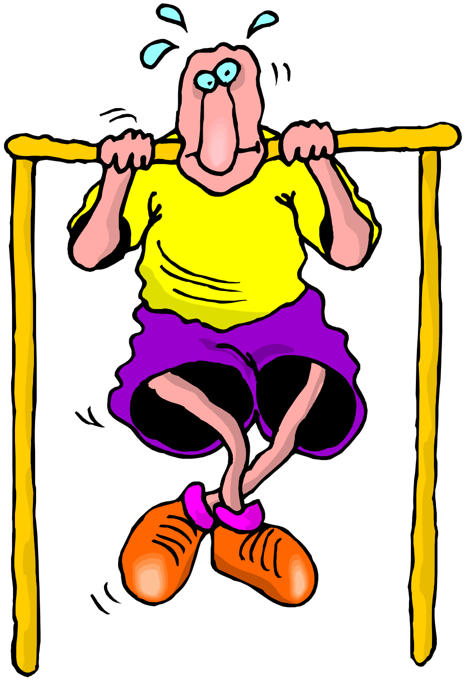
- GA-01 - Trial Balance Report
- GA-02 - Open Items Report
- GA-03 - Detailed Transaction Listing
- GA-04 - Detailed Fund Receipt
- GA-05 - Revenue Source Summary Report
- GA-06 - Summary of Activities Report
- GA-07 - Cash Balance Report
- GA-08 - Revenue Report
- GA-09 - Funds Balance Sheet Report
- GA-10 - Detailed Listing of Obligations Report

Import/Export Universes

- Import Universe: to get latest version from Repository
  - File -> Import
  - Double-clicking will lock universe, preventing others to export universe
- Export Universe: to share changes with other users (if not exported, changes remain local)
  - File -> Export
- File location where local copies are stored: Tools -> Options -> Save

Exercise / Demo

- Open AR Universe
- Browse tables
- Explore Objects



Developing Universes

Kentucky

UNBRIDLED SPIRIT™

Universe Development Process

- Prepare – Know the data source
- Analyze – Identify the needs
- Plan – Design the universe
- Implement – Create the universe
- Test – Using a test team and end users
- Deploy – Distribute the universe
- Evolve – Adapt and extend universe

Universe Development Process – Cont'd

- Prepare
  - Identify the target data source and become familiar with its structure
  - Know what data is contained within each table of the target database
  - Understand the joins
  - Identify the cardinality

Universe Development Process – Cont'd

- Analyze
  - Identify the user population and how it is structured
  - Identify what information the users need
  - Identify what standard reports they require
  - Familiarize yourself with business terminology to name objects
- Plan
  - Identify a strategy (how many universes, linking, etc.)
    - Linked universes: Fields that are commonly used can be in one universe that other universes link in
      - Easier maintenance
      - Consistency

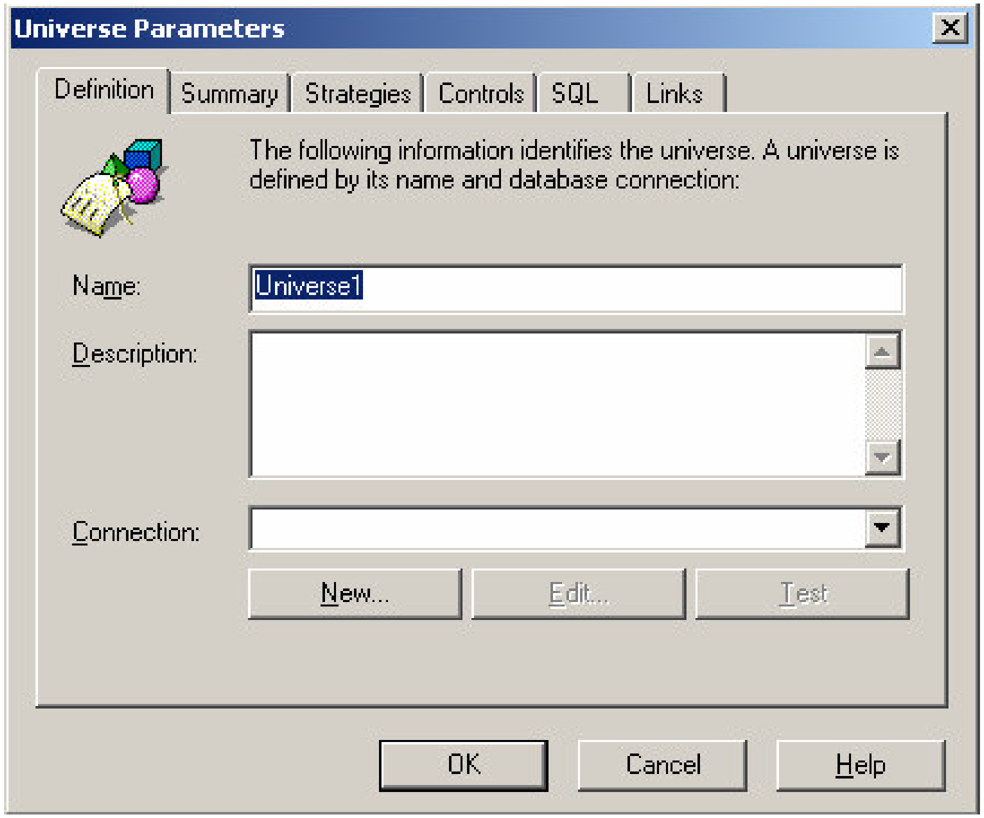
Universe Development Process – Cont’d

- Implement
  - Build the universe using Designer
  - Test frequently during the build process
- Test
  - Form a small group of users who know what information to expect from the universe
  - Test reports
- Deploy
  - Distribute the universe via the repository
- Evolve
  - Update and maintain the universe as the data sources and requirements change

Universe Parameters

Initial set up when creating a new universe:

- Name the Universe
- Select a connection
  (TEST !)
- Parameter comments
- Limit number of rows returned
- Limit execution time
- Cartesian Products: prevent or warning
- File -> Parameter to update settings later



The following information identifies the universe. A universe is defined by its name and database connection:

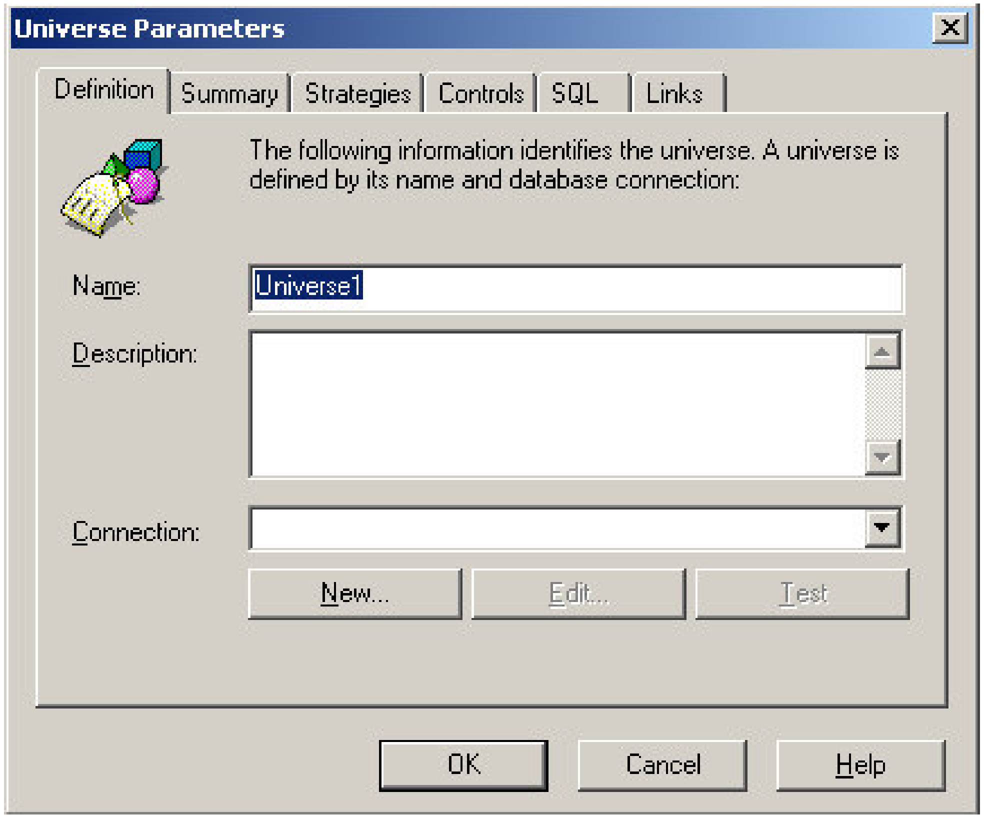
Name: Universe1

Description:

Connection:

Universe Parameters - Definition

- Includes Universe name, Description and Connection information



Universe Parameters

The following information identifies the universe. A universe is defined by its name and database connection:

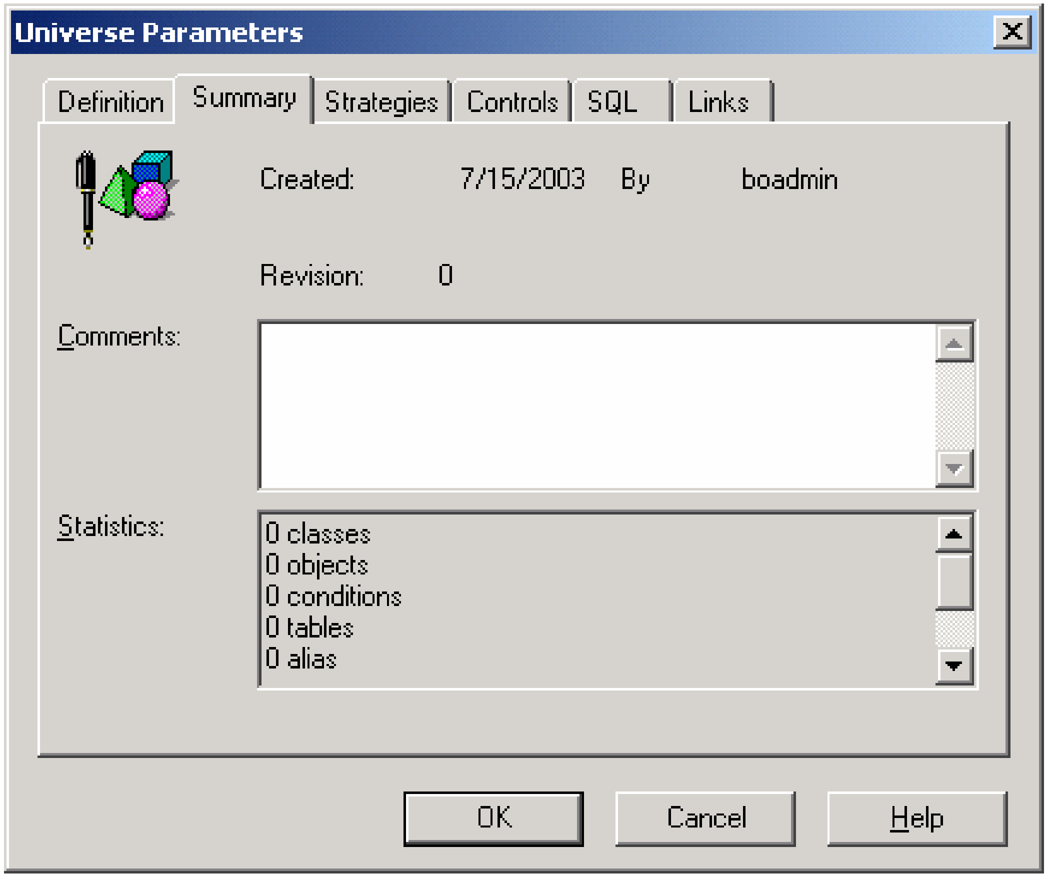
Name: Universe1

Description:

Connection:

Universe Parameters - Summary

- Includes Universe statistic information



Universe Parameters

Created: 7/15/2003 By boadmin

Revision: 0

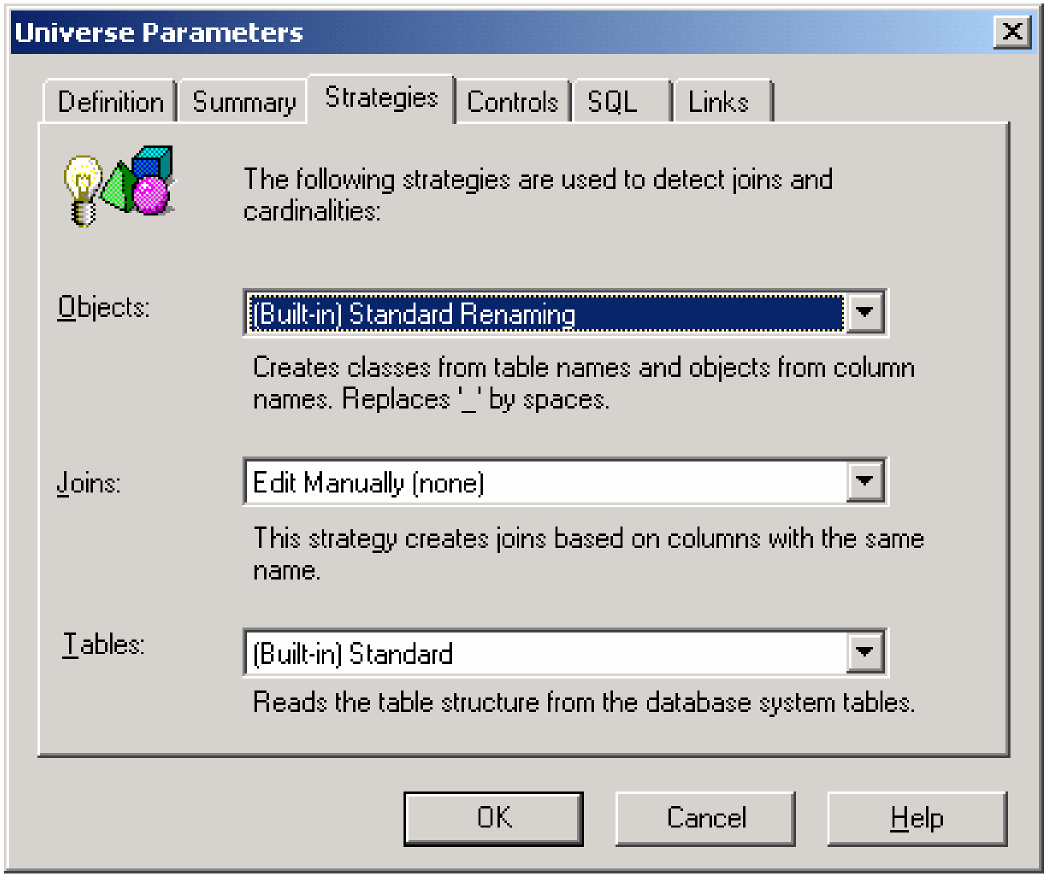
Comments:

Statistics:

0 classes
0 objects
0 conditions
0 tables
0 alias

Universe Parameters - Strategies

- Includes options for changing the approach to creating objects and tables
- Normally do not change



The following strategies are used to detect joins and cardinalities:

Objects: (Built-in) Standard Renaming

Creates classes from table names and objects from column names. Replaces '_' by spaces.

Joins: Edit Manually (none)

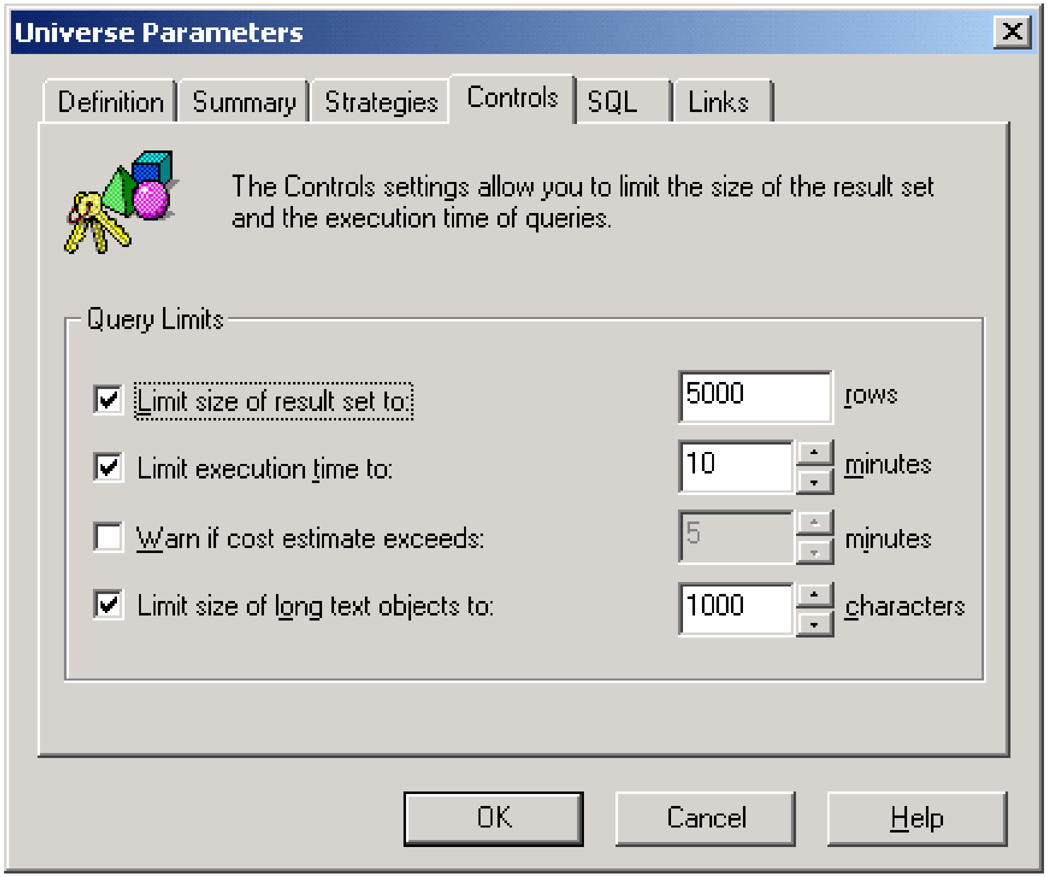
This strategy creates joins based on columns with the same name.

Tables: (Built-in) Standard

Reads the table structure from the database system tables.

Universe Parameters - Controls

- Includes Universe data and time restrictions
- Do not limit the size of result set
- Do not limit execution time or Significantly increase number of minutes



Universe Parameters

The Controls settings allow you to limit the size of the result set and the execution time of queries.

Query Limits

☑ Limit size of result set to: 5000 rows

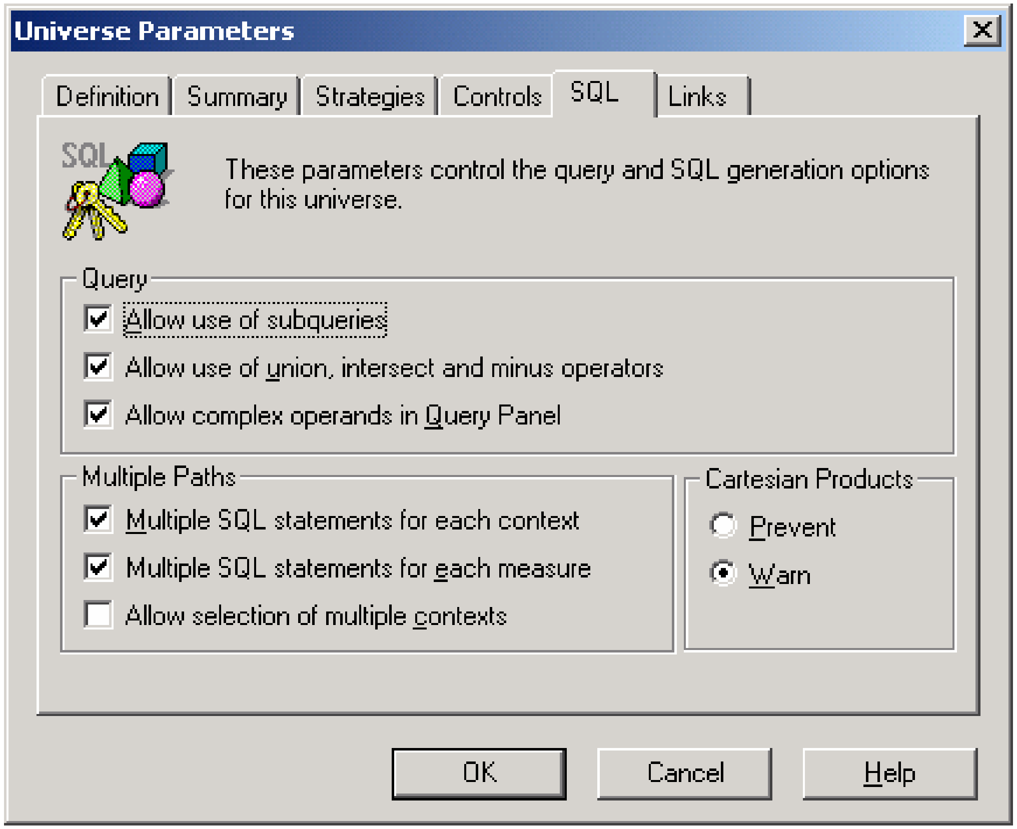
☑ Limit execution time to: 10 minutes

□ Warn if cost estimate exceeds: 5 minutes

☑ Limit size of long text objects to: 1000 characters

Universe Parameters - SQL

- Controls the SQL generation options for a universe



These parameters control the query and SQL generation options for this universe.

Query

☑ Allow use of subqueries

☑ Allow use of union, intersect and minus operators

☑ Allow complex operands in Query Panel

Multiple Paths

☑ Multiple SQL statements for each context

☑ Multiple SQL statements for each measure

□ Allow selection of multiple contexts

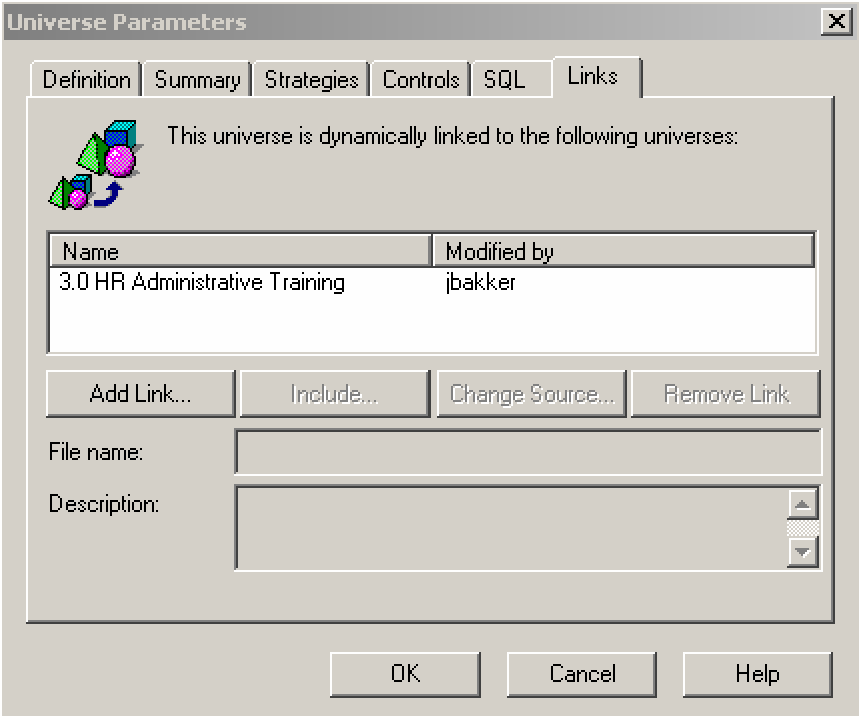
Cartesian Products

○ Prevent

● Warn

Universe Parameters – Links

- Universes can be linked
- Convenient for tables used in many universes:
  - COA
  - Location
- Linked in universes should be exported first



This universe is dynamically linked to the following universes:

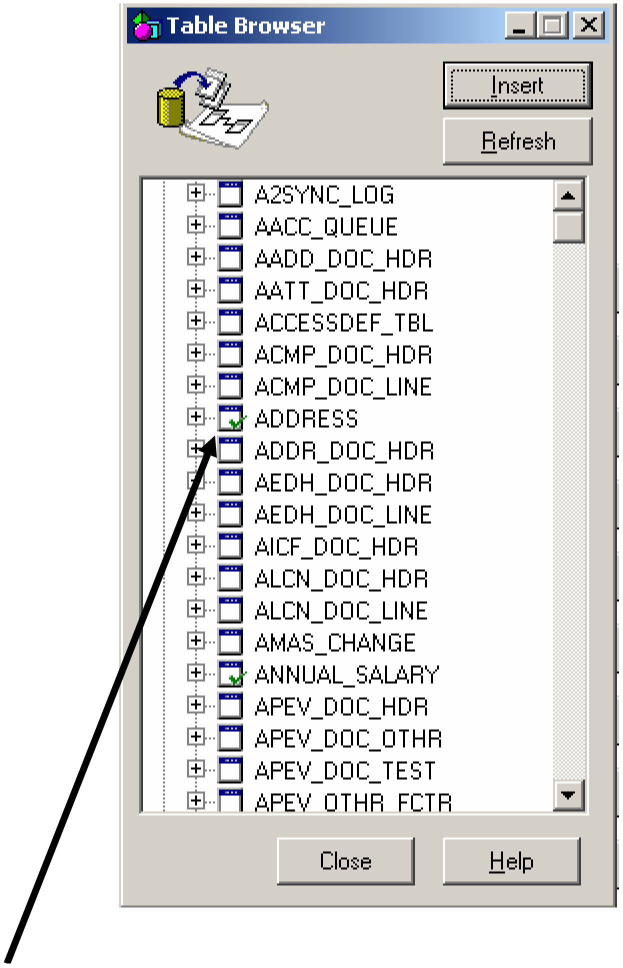
Name	Modified by
3.0 HR Administrative Training	jbakker

File name:

Description:

Adding Tables

- Tables can be added to each universe
- Table needs to be present in universe to be used in reports
- Right mouse click, double click or ‘Insert – Tables’ will work
- Create objects/classes to allow use in reports
- Tables are added in right hand pane



A2SYNC_LOG
AACC_QUEUE
AADD_DOC_HDR
AATT_DOC_HDR
ACCESSDEF_TBL
ACMP_DOC_HDR
ACMP_DOC_LINE
✓ ADDRESS
ADDR_DOC_HDR
AEDH_DOC_HDR
AEDH_DOC_LINE
AICF_DOC_HDR
ALCN_DOC_HDR
ALCN_DOC_LINE
AMAS_CHANGE
✓ ANNUAL_SALARY
APEV_DOC_HDR
APEV_DOC_OTHR
APEV_DOC_TEST
APEV_OTHR_FCTR

✓ Already in Universe

Adding Tables

- Everything in the right hand panel is invisible to the end user
- If universe was saved and exported after only adding tables, there would be no objects or classes to use in creating a report

Exercise

Create New Universe

- Create a new Universe
- Name the universe [LAST NAME] Journal Accounting
- Select the DataWarehouse connection

Setup Universe Parameters

- Remove Size of result set limit
- Remove Execution time limit

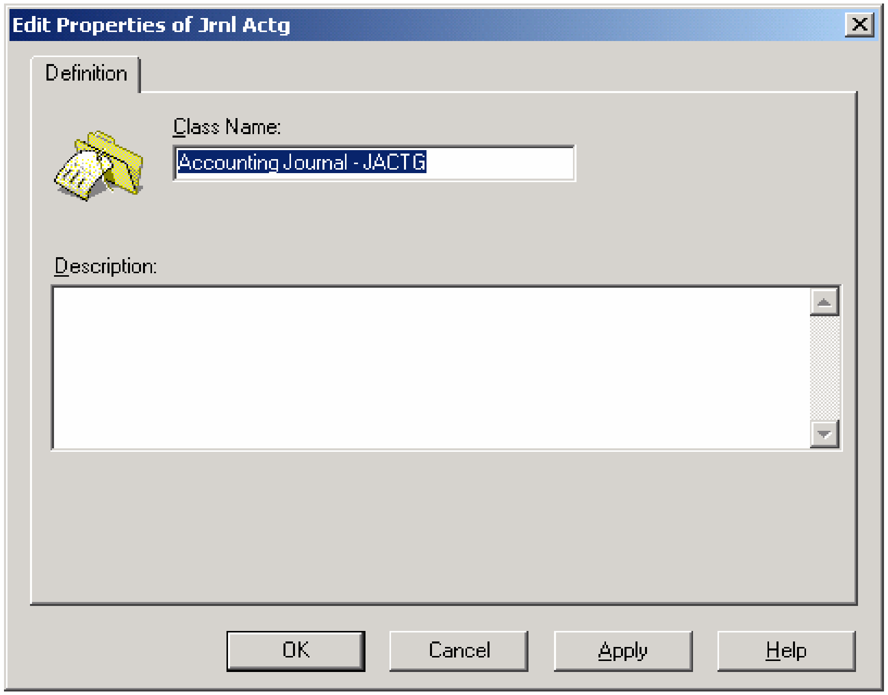
Add tables to Universe

- Add the following tables:
  - FACT_JRNL_ACTG
  - DIM_FY
  - DIM_FUND
  - DIM_ORG
- Save your Universe

Classes and Objects

Classes and Objects

- Classes and Objects are equivalent to tables and rows
- Define Class Properties
  - Class properties should be changed for each class
  - Give class a more recognizable name
  - Add a description if necessary



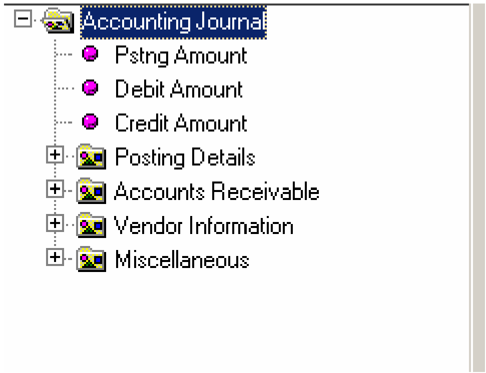
Edit Properties of Jrnl Actg

Class Name: Accounting Journal - JACTG

Description:

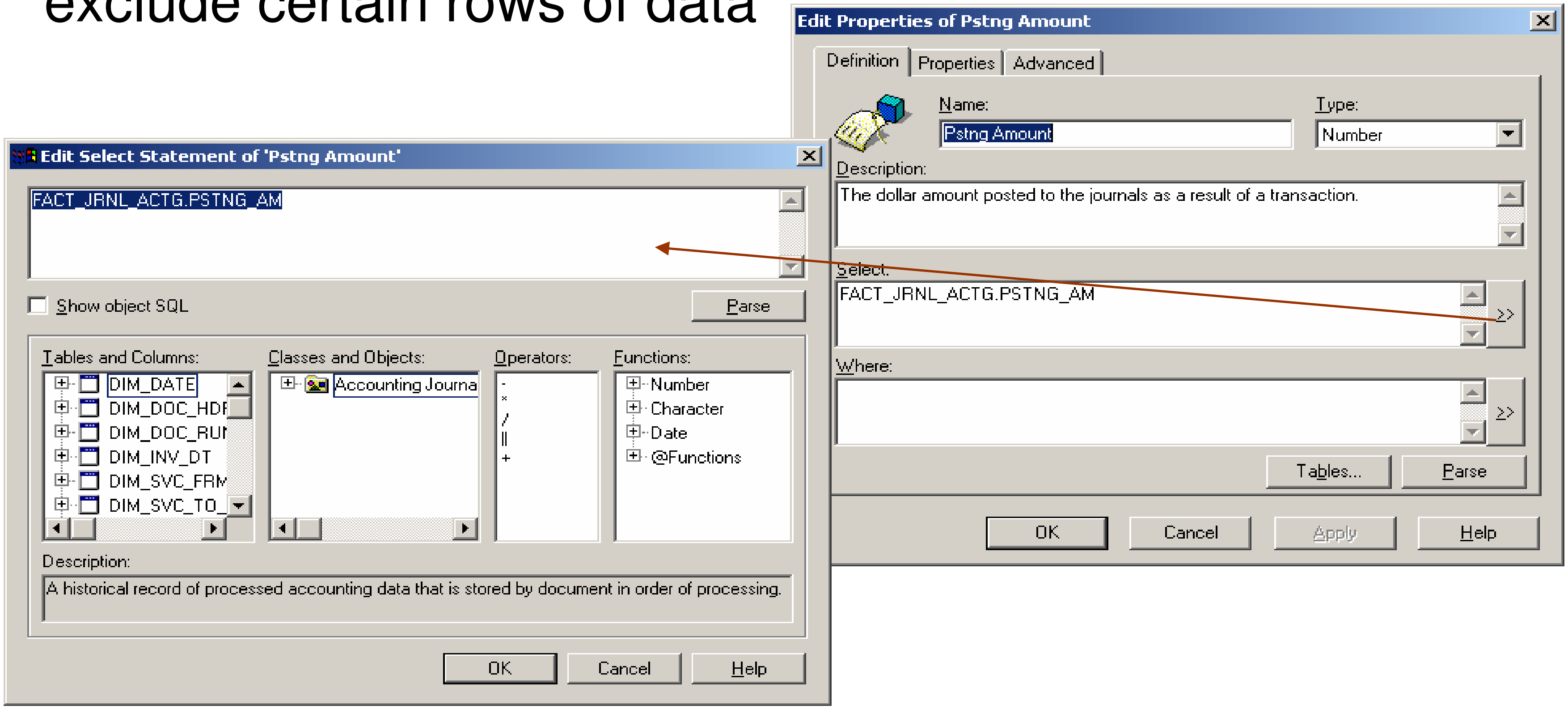
Classes and Objects

- Subclasses are classes within classes
- Subclasses are used to organize Classes and Objects
- Can only use subclass name once in the Universe
- Add a description if necessary
- Subclasses within a Subclass is allowed



Object Definition

- Select field defines from where data is inferred
- Where field is a clause that is used to either include or exclude certain rows of data



Edit Properties of Pstng Amount

Name: Pstng Amount

Type: Number

Description: The dollar amount posted to the journals as a result of a transaction.

Select: FACT_JRNL_ACTG.PSTNG_AM

Where:

Edit Select Statement of 'Pstng Amount'

FACT_JRNL_ACTG.PSTNG_AM

□ Show object SQL

Tables and Columns:
DIM_DATE
DIM_DOC_HDF
DIM_DOC_RUI
DIM_INV_DT
DIM_SVC_FRM
DIM_SVC_TO_

Classes and Objects:
Accounting Journa

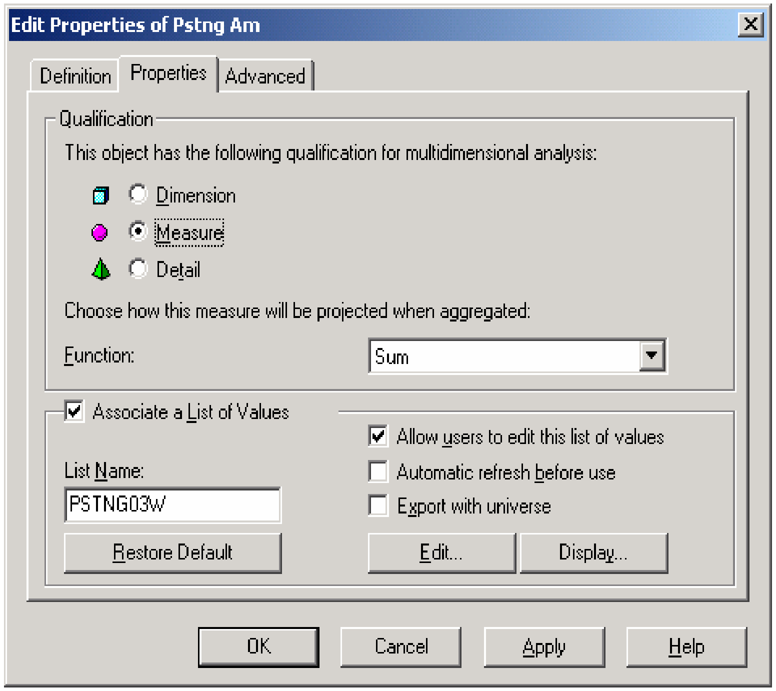
Operators:
-
*
/
||
+

Functions:
Number
Character
Date
@Functions

Description:
A historical record of processed accounting data that is stored by document in order of processing.

Object Properties

- Object qualification
  - Dimension: Typically a character field; it is the default
  - Measure: Numeric fields; can be aggregated
  - Detail: Information about dimension



Edit Properties of Pstng Am

Qualification

This object has the following qualification for multidimensional analysis:

○ Dimension
● Measure
○ Detail

Choose how this measure will be projected when aggregated:

Function: Sum

☑ Associate a List of Values

List Name: PSTNG03W

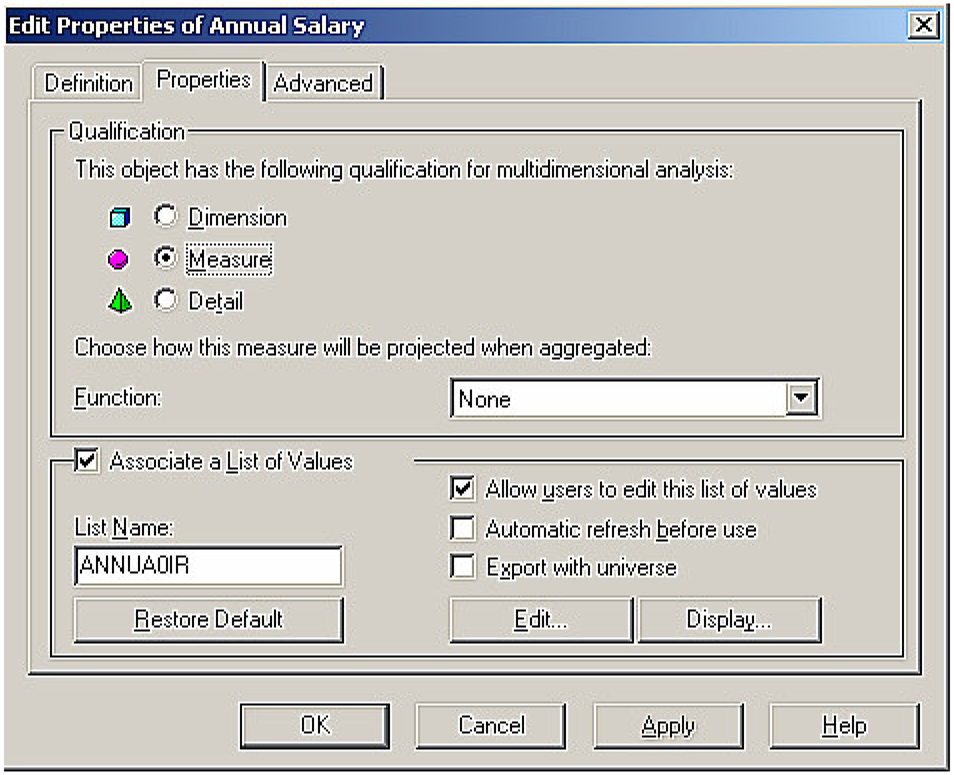
☑ Allow users to edit this list of values
□ Automatic refresh before use
□ Export with universe

Measure Objects

- Returns numeric information
- Created by using aggregate functions
- 5 basic aggregate functions
  - Sum
  - Count
  - Average
  - Maximum
  - Minimum
- Infers a “Group By” clause in the Select statement if it also includes other types of objects

Creating (Measure) Objects

- Data Type must be set to Number
- The Select field must only contain an aggregate
- Qualification must be set as a Measure type and a projected aggregate function must be set in the Function field
- A list of values should NOT be associated with the object



Edit Properties of Annual Salary

Qualification

This object has the following qualification for multidimensional analysis:

○ Dimension
● Measure
○ Detail

Choose how this measure will be projected when aggregated:

Function: None

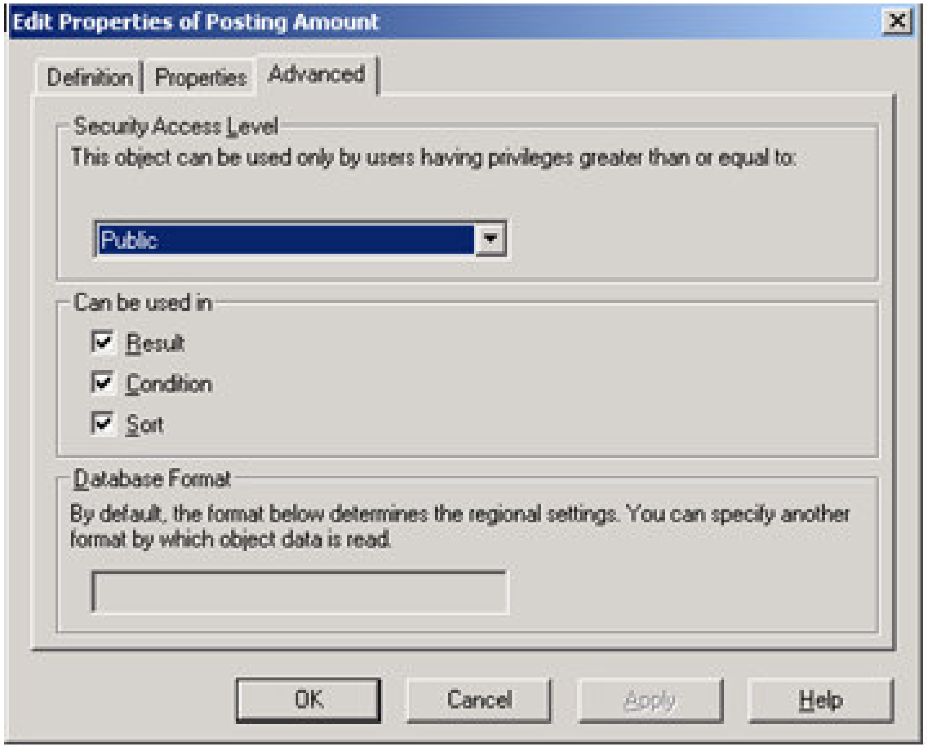
☑ Associate a List of Values

☑ Allow users to edit this list of values
□ Automatic refresh before use
□ Export with universe

List Name: ANNUA0IR

Creating (Measure) Objects – Cont'd

- Security can be set at this level;
  - Define where object can be used
  - Define who can use it (security role)
- Beware of maintenance issues



Edit Properties of Posting Amount

Security Access Level

This object can be used only by users having privileges greater than or equal to:

Public

Can be used in

☑ Result

☑ Condition

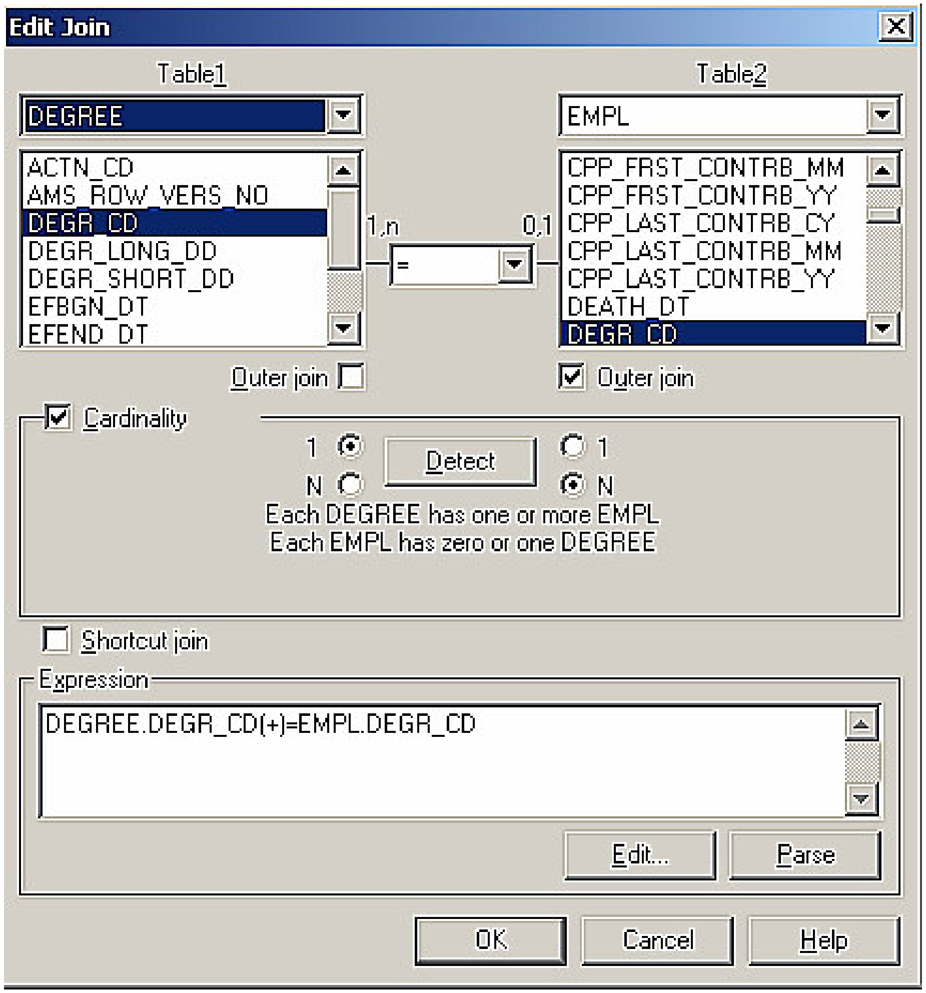
☑ Sort

Database Format

By default, the format below determines the regional settings. You can specify another format by which object data is read.

Table Joins

- Tables need to be joined in order to use them in reports
- Needed to prevent Cartesian products
- Joins restrict the result set of a multi-relational query
- 4 types of joins in AMS *info*Advantage
  - Equi-joins
  - Outer-joins
  - Theta Joins
  - Self – restricting joins



Edit Join

Table1: DEGREE

ACTN_CD
AMS_ROW_VERS_NO
DEGR_CD
DEGR_LONG_DD
DEGR_SHORT_DD
EFBGN_DT
EFEND_DT

Table2: EMPL

CPP_FRST_CONTRB_MM
CPP_FRST_CONTRB_YY
CPP_LAST_CONTRB_CY
CPP_LAST_CONTRB_MM
CPP_LAST_CONTRB_YY
DEATH_DT
DEGR_CD

1,n = 0,1

□ Outer join ☑ Outer join

☑ Cardinality

1 ● ○ 1
N ○ ● N

Each DEGREE has one or more EMPL
Each EMPL has zero or one DEGREE

□ Shortcut join

Expression

DEGREE.DEGR_CD(+)=EMPL.DEGR_CD

Kentucky UNBRIDLED SPIRIT

eMARS Management Administrative & Reporting System

Joins – 1

Equi Join:

- A join that conforms to the following syntax:
  - Table1.column_a = Table2.column_a
- Based on column values
- In a normalized table, usually the primary key
- The where clause applies a restriction to provide only matched data to be returned

Theta Joins:

- Links tables based on a relationship other than equality

Joins – 2

Outer Join

- Join for the following situation: ‘Table A contains values for records that don’t have matching values/records in table B’
- Specify which table is the outer join table
- Shown by a small circle on the join line

Cardinalities

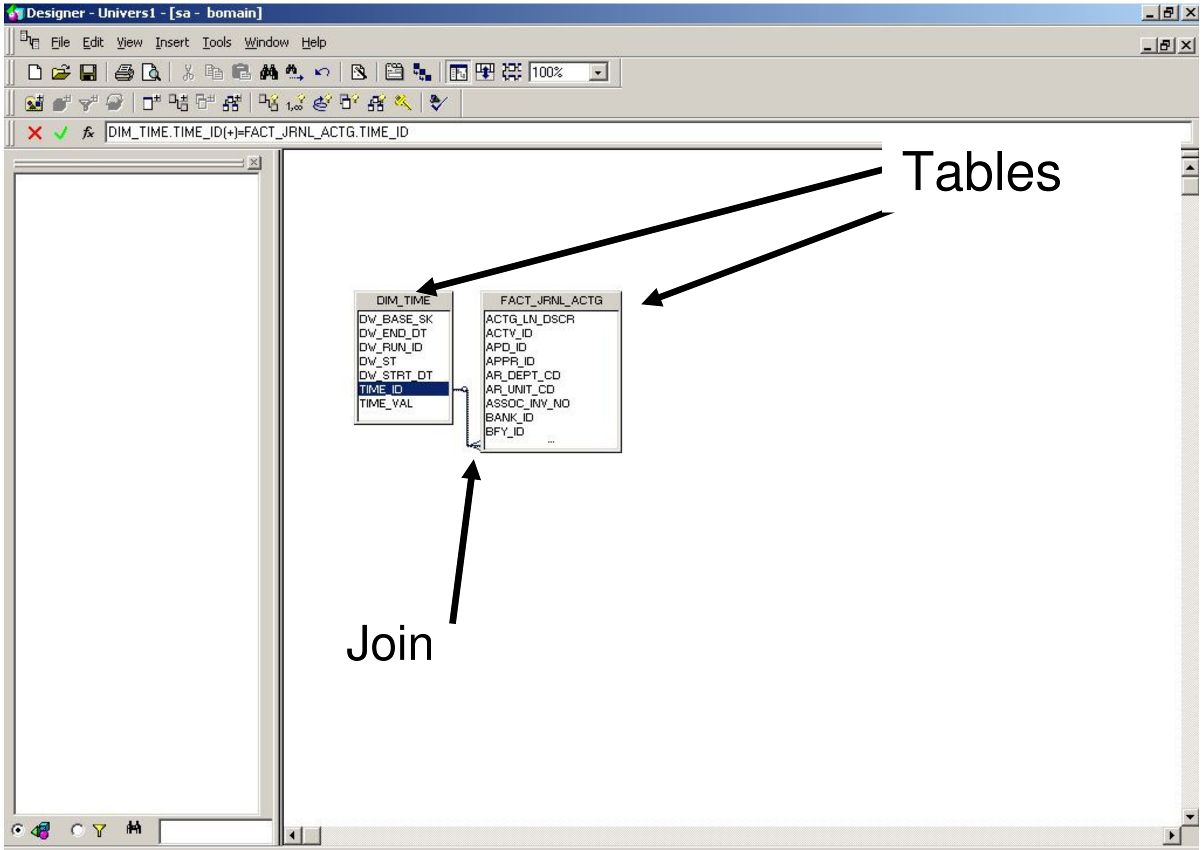
- Term used for expressing the relationship between tables based on joins
  - Zero or One to Many
  - Zero or One to One
  - Many to One or Zero
- Does not impact SQL, used to detect and resolve loops
- Can be a useful aid in understanding database structures
- Do **NOT** <u>automatically</u> <u>detect</u> when using large database

Checking Integrity

- Used to detect any errors in the structure, joins or cardinalities of a universe
- Careful on checking cardinality with a large database
- Tools -> Check Integrity

Create Objects & Classes

- Tables are added in right hand panel
- Joins are defined there too



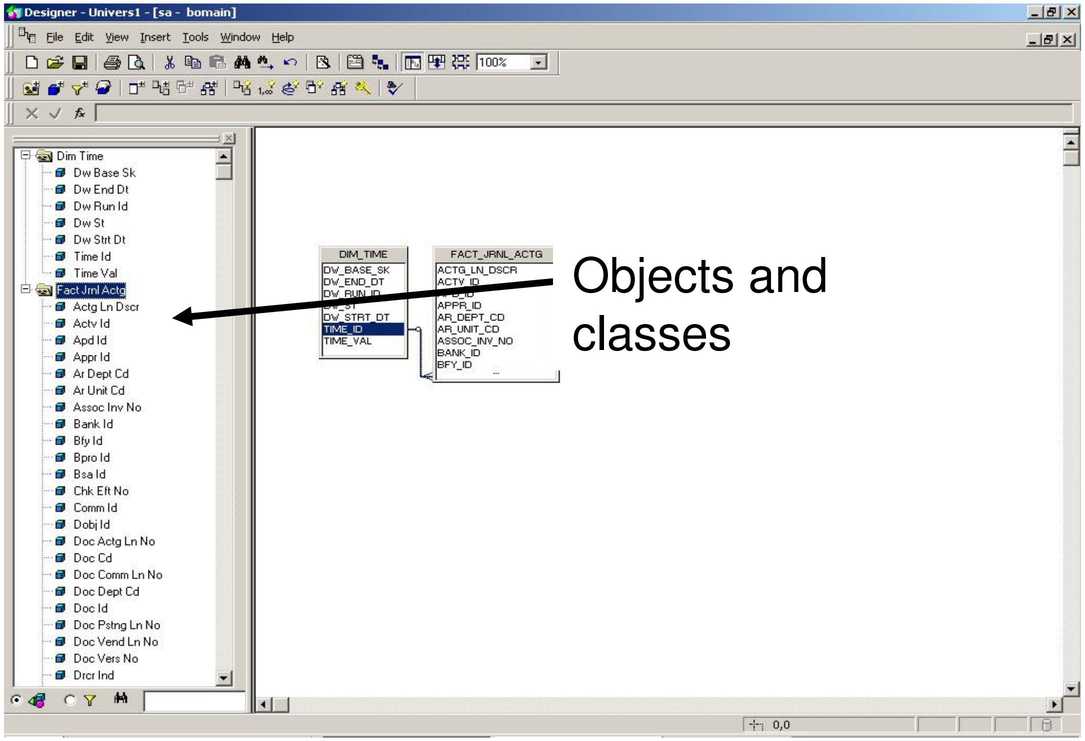
DIM_TIME.TIME_ID(+)=FACT_JRNL_ACTG.TIME_ID

Tables

Join

Create Objects and Classes – 2

- Objects and classes need to be added to left hand side in order to be used in reporting
- Drag and drop entire table



Objects and classes

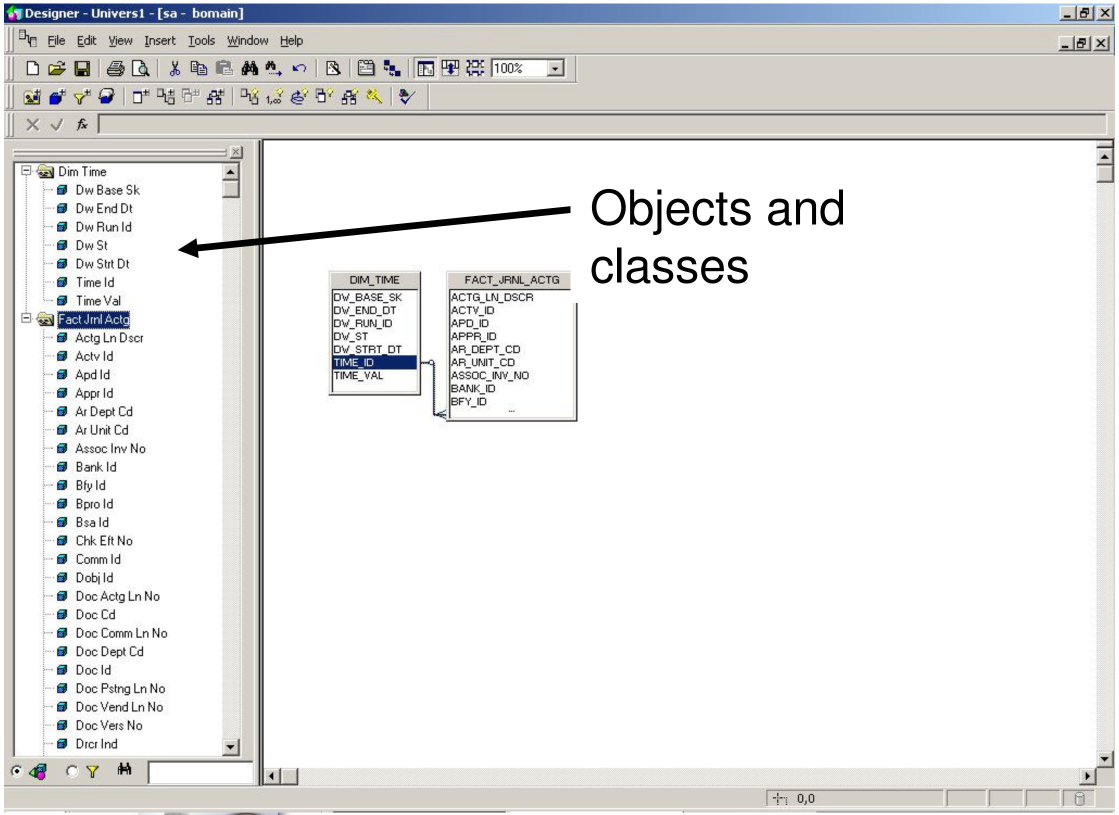
- Table = Class
- Field = Object
  - Default is dimension

OR

- Add classes / objects manually in left hand panel

Create Objects and Classes – 2

- Objects and classes need to be added to left hand side in order to be used in reporting
- Drag and drop entire table



Objects and classes

- Table = Class
- Field = Object
  - Default is dimension

OR

- Add classes / objects manually in left hand panel

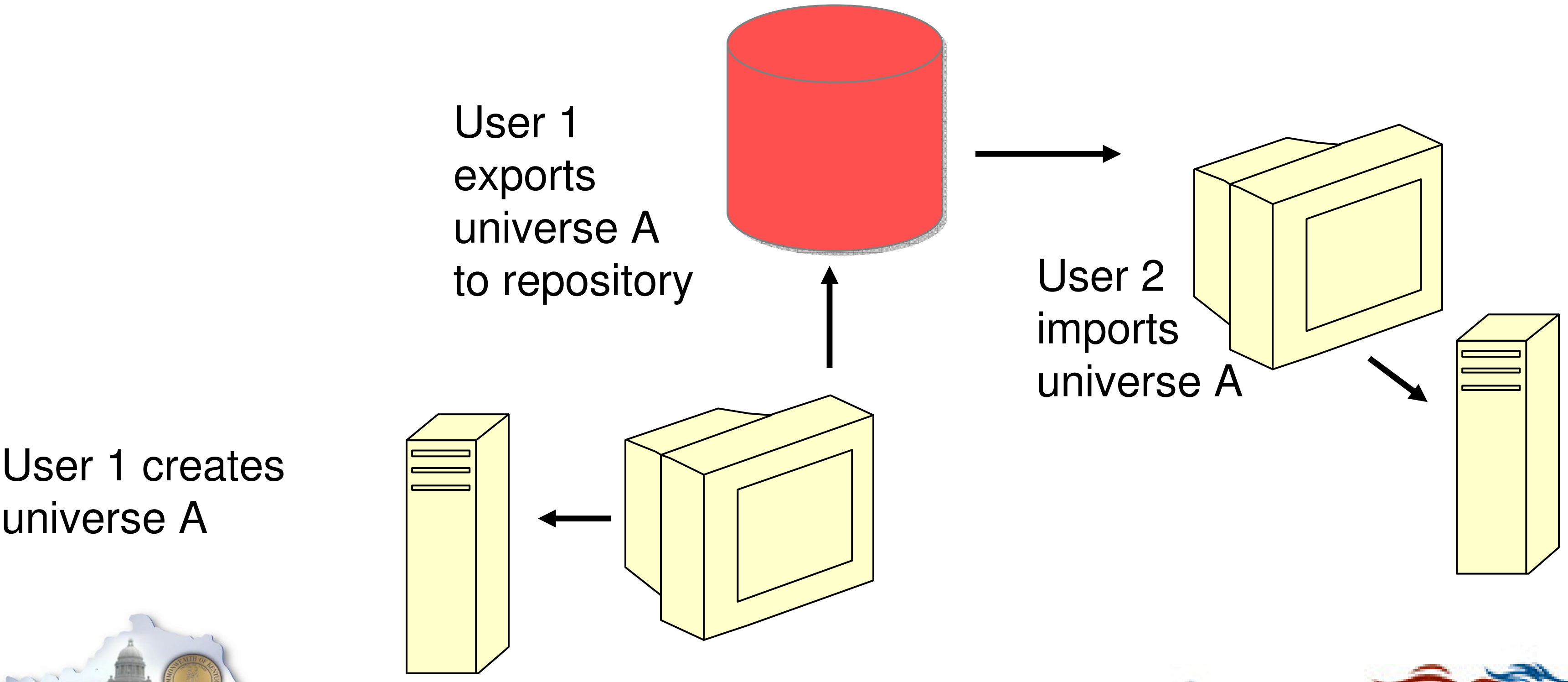
Exercise

- Open your saved universe
  - Create a class called Journal Accounting
    - Include the following fields from the FACT_JRNL_ACCTG table and rename the objects as follows
      - Func ID -> Function ID
      - Pstng Am -> Posting Amount
      - Rec No -> Record Number
      - Jrnl Type -> Journal Type
      - Time Id -> Time id
    - Change Posting Amount to a measure object with no object function
    - Copy Posting Amount and call it Summed Posting Amount. Make this new object a measure with a Sum Object Function
  - Create a sub class under Journal Accounting called Fund
    - Include the following fields from the DIM_FUND table and rename the objects as follows
      - Fund id -> Fund Id
      - Fund Sh Nm -> Fund Short Name
      - Fund Nm -> Fund Name
    - Change the Fund Short Name and Fund Name to Detail Objects associated with Fund
  - Create a Class called Fiscal Year
    - Include an object for every field on the DIM_FY table
  - Hide the Time Id Object in the Journal Accounting Class
  - Create a new object of type measure named Posting Amount Mod so that it is equal to Posting Amount – Posting Amount.

Universe Management

Universe Actions

- When developing/modifying a universe on a local machine, the new/changed universe is not available to other users
- Need to export to the repository so others can see it



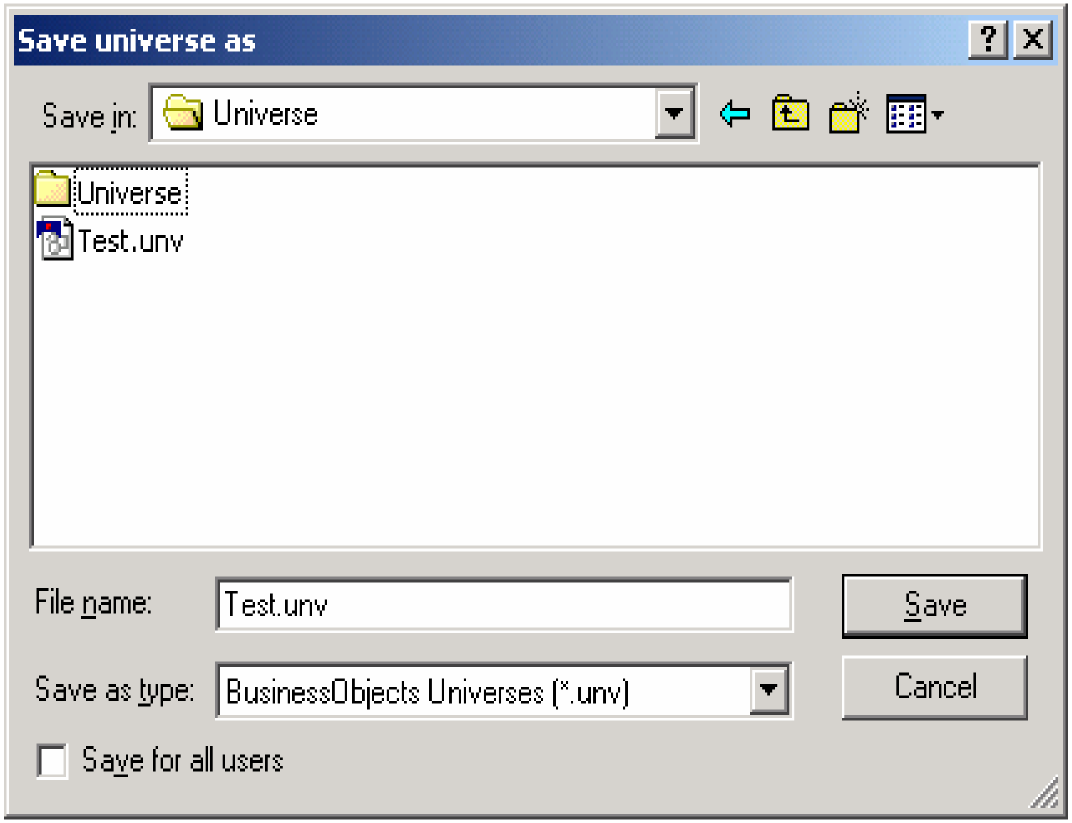
User 1 creates universe A

User 1 exports universe A to repository

User 2 imports universe A

Saving a Universe

- File -> Save
- File Name cannot be longer than eight characters
- By default saved in the Universe subfolder in the Business Objects path



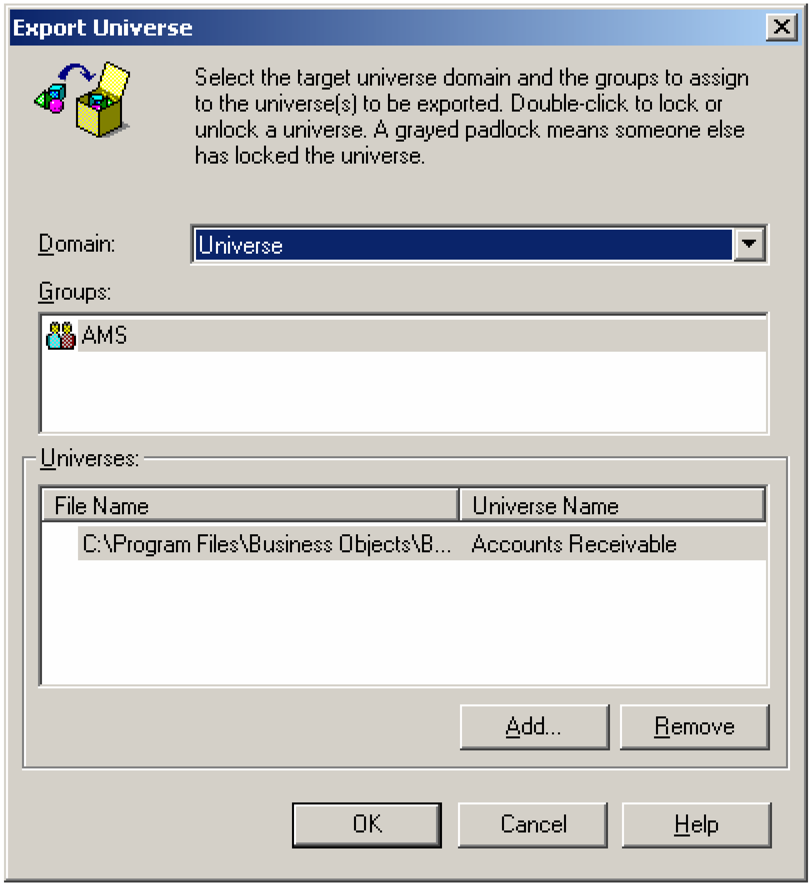
File name: Test.unv

Save as type: BusinessObjects Universes (*.unv)

□ Save for all users

Exporting a Universe

- By default, program will export Universes that are open in Designer
- Can export more than one Universe at a time
- If multiple Universe domains exist, choose from drop down menu



Export Universe

Select the target universe domain and the groups to assign to the universe(s) to be exported. Double-click to lock or unlock a universe. A grayed padlock means someone else has locked the universe.

Domain: Universe

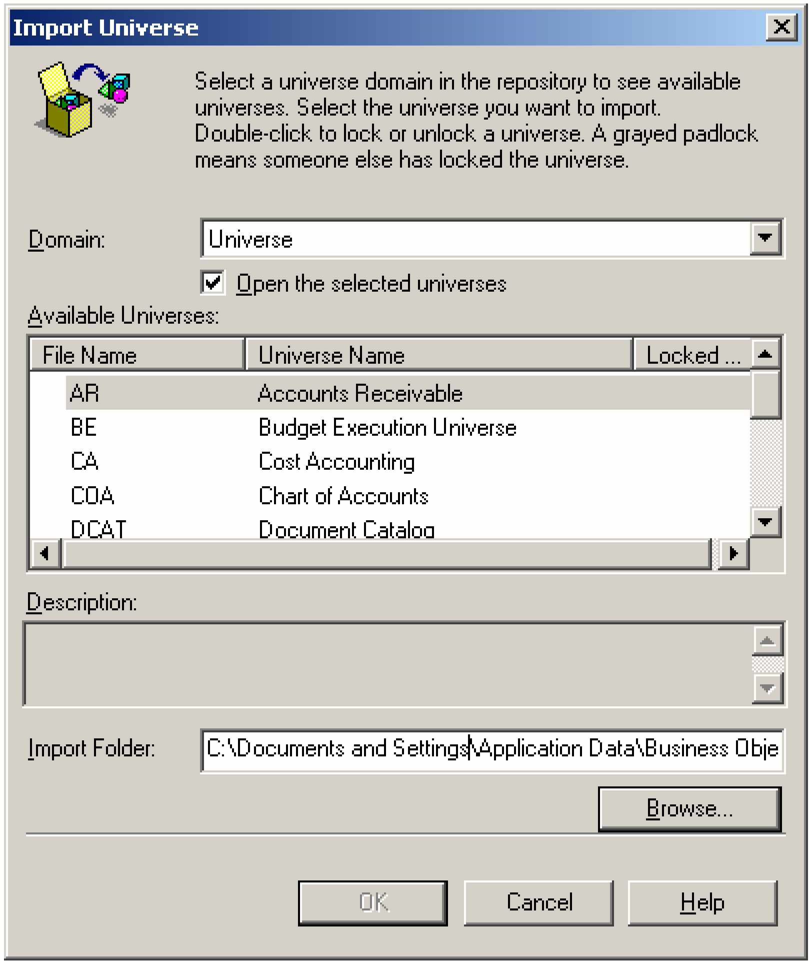
Groups: AMS

Universes:

File Name	Universe Name
C:\Program Files\Business Objects\B...	Accounts Receivable

Importing a Universe

- To modify Universes that have been exported, import the Universe for the most recent copy
- Will create a local copy
- Can import more than one Universe at a time
- If multiple Universe domains exist, choose from drop down menu
- Double clicking will lock Universe, preventing others from exporting the same Universe



Import Universe

Select a universe domain in the repository to see available universes. Select the universe you want to import. Double-click to lock or unlock a universe. A grayed padlock means someone else has locked the universe.

Domain: Universe

☑ Open the selected universes

Available Universes:

File Name	Universe Name	Locked ...
AR	Accounts Receivable	
BE	Budget Execution Universe	
CA	Cost Accounting	
COA	Chart of Accounts	
DCAT	Document Catalog	

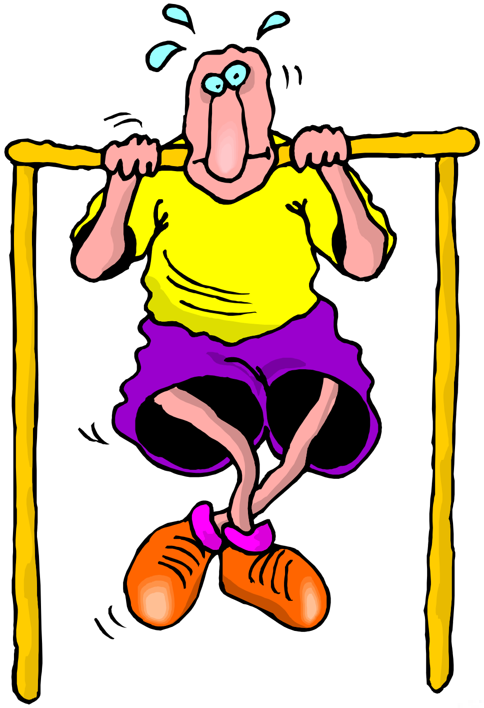
Description:

Import Folder: C:\Documents and Settings\Application Data\Business Obje

Exercise

Create New Universe

- Create a new Universe
- Name the universe [LAST NAME] Journal Accounting
- Select the DataWarehouse connection
  - TEST connection
- **Setup Universe Parameters (Control)**
  - Remove Size of result set limit
  - Remove Execution time limit
- **Add following tables to Universe**
  - FACT_JRNL_ACTG
  - DIM_TIME
- Save your Universe

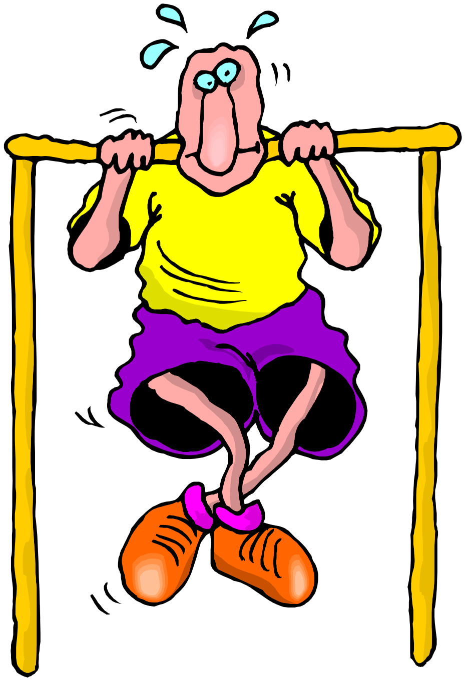


Exercise Cont'd

- Add Outer Join
  - Time_ID object in both tables
  - Cardinality: Each Dim_Time has 1 or more FACT_JRNL_ACTG
  - Each FACT_JRNL_ACTG has zero or one DIM_TIME
- Save Universe
- Open Business Objects ('Thick Client')
  - Create report with Time_ID object using the new Universe
  - **WHAT HAPPENED?**

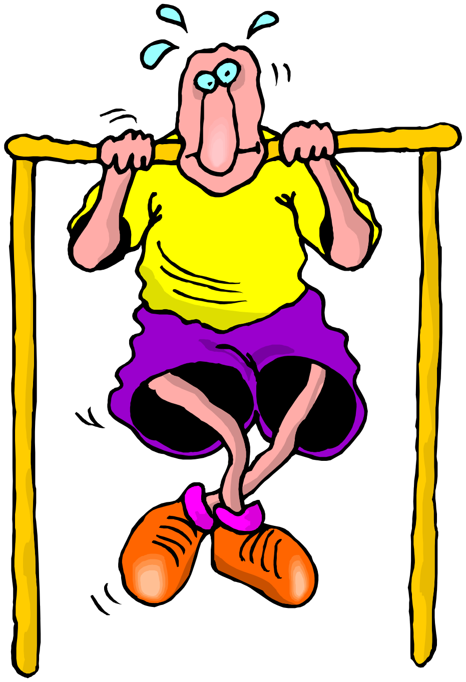
Exercise Cont'd

- Open newly created Universe
  - Add classes and objects
  - SAVE
- Switch back to 'Thick Client'
  - Create report with Time_ID object using the new Universe
- Open Webintelligence
- Create new report with newly created Universe
  - **WHAT HAPPENED? WHY?**



Exercise – Cont'd

- Export Universe to repository
- Return to Webintelligence
- Create report using new universe
- Extra: try to add more tables
  - Note: for relationships see other universes
  - Add New Object, not directly a map to a DB field (example measure object with additional formula)



Questions?